

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

*Кафедра автоматизованих систем обробки інформації та управління*

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_  
(підпис) Олександр ПАВЛОВ  
(вл.ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інформаційні управляючі  
системи та технології»  
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

**на тему: *Інформаційна система з підтримки інвестиційно-  
фінансової діяльності***

**Виконав :**

студент IV курсу, групи ІС-61

Ходарченко Андрій Андрійович

(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Керівник**

ст.вик. Ковтунець Олесь Володимирович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Консультант з  
графічної  
документації**

доц., к.т.н., доц. Тєлишева Тамара Олексіївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

**Рецензент**

доц., к.т.н., доц. Пасько Віктор Петрович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент (-ка) \_\_\_\_\_  
(підпис)

Київ – 2020 року

**Національний технічний університет України**  
**“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ  
(підпис) (вл.ім'я, прізвище)

“ ” 2020 р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**

Ходарченко Андрій Андрійович  
(прізвище, ім'я, по батькові)

1. Тема проєкту «*Інформаційна система з підтримки інвестиційно-фінансової діяльності*»

керівник проєкту Ковтунець Олександр Володимирович, ст. вик  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7”травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01”червня 2020 року

3. Вихідні дані до проєкту

*Технічне завдання*

4. Зміст пояснювальної записки

*1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі*

*2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних*

*3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання*

*4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення*

*5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту*

## 5. Перелік графічного матеріалу

1. *Схема структурна діаграми послідовності*

2. *Схема структурна класів програмного забезпечення*

3. *Схема структурна діаграми розгортання*

4. *Схема бази даних*

5. *Креслення екранних форм*

## 6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

## 7. Дата видачі завдання «13» квітня 2020 року

### Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>10.02.2020</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>15.03.2020</i>	
3.	<i>Підготовка технічного завдання на розробку програмного продукту</i>	<i>17.03.2020</i>	
4.	<i>Розробка сценарію роботи</i>	<i>24.03.2020</i>	
5.	<i>Технічне проектування – функціональність, модулі, задачі, цілі тощо</i>	<i>30.03.2020</i>	
6.	<i>Узгодження з керівником інтерфейсу користувача</i>	<i>15.04.2020</i>	
7.	<i>Розробка інформаційного забезпечення</i>	<i>30.04.2020</i>	
8.	<i>Розробка програмного забезпечення</i>	<i>05.05.2020</i>	
9.	<i>Налагодження програми</i>	<i>10.05.2020</i>	
10.	<i>Тестування програми</i>	<i>27.05.2020</i>	
11.	<i>Оформлення пояснювальної записки</i>	<i>30.05.2020</i>	
12.	<i>Подання ДП на попередній захист</i>	<i>15.05.2020</i>	
13.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
14.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Андрій ХОДАРЧЕНКО

Керівник

Олесь КОВТУНЕЦЬ

[illegible]

# **Пояснювальна записка до дипломного проекту**

на тему: Інформаційна система з підтримки інвестиційно-фінансової  
діяльності

---

Київ – 2020 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проекту складається з шести розділів, містить 19 рисунків, 10 таблиць, 1 додаток, 13 джерел.

Дипломний проект присвячений розробці тренажеру з рекомендаціями для освоєння процесу інвестування.

У розділі інформаційного забезпечення описана структура вхідних та вихідних даних, наведена структура бази даних.

Розділ математичного забезпечення присвячений математичній постановці задачі та методу її вирішення.

Програмне забезпечення описане у розділі «Програмне та технічне забезпечення». Тут описані основні програмні засоби, які були використані для розробки даної програми, наведені технічні вимоги до системи, на якій буде запускатися програма, описана програмна архітектура, яка була обрана для розробки.

У технологічному розділі наведено скріншоти програми, проведено функціональні тести для перевірки відповідності програмного продукту до технічного завдання.

АКЦІЇ, ВАЛЮТА, РЕКОМЕНДАЦІЇ, СИМПЛЕКС МЕТОД,  
ТРЕНАЖЕР, ІНВЕСТУВАННЯ.

					ДП ІС-6127.00.000 ПЗ				
		Прізвище	Підпис	Дата					
Розроб.		Ходарченко А.А			Інформаційна система з підтримки інвестиційно- фінансової діяльності	Літ.		Лист	Листів
Перевірив.		Ковтунець О.В						2	
Н. кон.		Тєлишева Т.О							
Затв.		Павлов О.А.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61			

## ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of six sections, contains 19 figures, 10 tables, 1 annex and 13 sources.

The diploma project is dedicated to the development of a simulator with recommendations for mastering the investment process.

In the information support section the structure of input and output data is described, and the structure of the database is given.

The section of mathematical support is devoted to mathematical problem definition and method of its solution.

The software is described in the section "Software and hardware". It describes the main software tools that were used for the development of this program, technical requirements to the system on which the program will be run, described program architecture that was selected for development.

In the technological section there are screenshots of the program, functional tests for checking compliance of the program product with the technical requirements.

STOCKS, CURRENCY, RECOMMENDATIONS, SIMPLEX  
METHOD, SIMULATOR, INVESTMENT.

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

## ЗМІСТ

ВСТУП .....	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	8
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА .....	8
1.1.1 Опис процесу діяльності .....	8
1.1.2 Опис функціональної моделі .....	8
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ .....	10
1.3 ПОСТАНОВКА ЗАДАЧІ .....	10
1.3.1 Призначення розробки .....	10
1.3.2 Цілі та задачі розробки .....	11
Висновок до розділу .....	11
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ .....	12
2.1 ВХІДНІ ДАНІ .....	12
2.2 ВИХІДНІ ДАНІ .....	12
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ .....	13
Висновок до розділу .....	15
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....	16
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ .....	16
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ .....	16
3.3 ОБГРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ .....	17
3.4 ОПИС МЕТОДІВ РОЗВ'ЯЗАННЯ .....	19
Висновок до розділу .....	19
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....	21
4.1 ЗАСОБИ РОЗРОБКИ .....	21
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	22
4.2.1 Загальні вимоги .....	22
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	23
4.3.1 Діаграма класів .....	23
4.3.2 Діаграма послідовності .....	23
4.3.3 Діаграма розгортання .....	23
4.3.4 Специфікація функцій .....	24



Висновок до розділу .....	25
<b>5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ .....</b>	<b>26</b>
5.1 КЕРІВНИЦТВО КОРИСТУВАЧА .....	26
5.2 ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	29
5.2.1 Мета випробувань .....	29
5.2.2 Загальні положення .....	29
5.2.3 Результати випробувань .....	29
Висновок до розділу .....	32
ЗАГАЛЬНІ ВИСНОВКИ .....	32
ПЕРЕЛІК ПОСИЛАНЬ .....	33
ДОДАТОК А .....	ERROR! BOOKMARK NOT DEFINED.

## ВСТУП

Інвестиція, капітальні вкладення — операція, що дозволяє інвестору купити певні активи. Це, наприклад, акції компаній, валюта іноземних держав, депозити, різні види матеріальних речей.

Інвестиційна діяльність — це процес придбання та продавання активів, менеджмент ризиків та прибутку інвестиційного портфеля, який супроводжується різними стратегіями. Інвестор прагне отримати прибуток.

Інвестиційна діяльність компанії - це послідовна цілеспрямована діяльність, яка полягає в капіталізації нерухомості, купівлі та використанні інвестиційних ресурсів, регулюванні інвестиційних процесів, міжнародному русі капіталу та інвестиційних товарів, а також у створенні сприятливих умов для інвестицій і мають на меті одержання прибутку.

Важливим поняттям в інвестиційній діяльності є інвестиційні цикли. Це процес, який реалізується протягом всього часу інвестування. Цикл існує від моменту формування намірів на придбання активу до моменту отримання прибутку від його реалізації.

Об'єктами в інвестиційної діяльності можуть виступати різні активи такі, як матеріальні цінності, цінні папери та майнові права, інтелектуальні цінності. Визначені види об'єктів інвестування можуть бути як об'єктами, в які вкладаються кошти, так і формами, в яких здійснюються вкладення.

Суб'єктами інвестиційної діяльності є інвестори та учасники, які можуть бути фізичними та юридичними особами.

Інвестори — суб'єкти інвестиційної діяльності. Вони приймають рішення про інвестування в актив, а саме про вкладання власних, позичкових і залучених майнових та інтелектуальних цінностей в об'єкти інвестування. Інвестори можуть бути вкладниками, кредиторами, покупцями.

За своїм статусом інвестори можуть бути корпоративними, інституційними та індивідуальними.

					ДП ІС-6127.00.000 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Індивідуальні або приватні інвестори – це фізичні особи, які можуть вкладати власні нагромадження для інвестування. Індивідуальними інвесторами може бути будь-хто. Індивідуальний інвестор найчастіше зацікавлений у отриманні прибутку з грошей, які він не використовує. Він прагне досягти фінансової стабільності.

Ризики інвестиційної діяльності – це можливість отримати відмінний від очікуваного результат. Це може статися з різних причин: закриття банків, компаній, криза і так далі. Очікуваний результат – це отримання доходів або досягнення певних цілей інвестора.

Інвестор для своєї діяльності використовую різні методи та стратегії. Процес інвестування зводиться до таких кроків:

- 1) Формування інвестиційного портфелю.
- 2) Вибір стратегії управління портфелем.
- 3) Редагування портфелю.
- 4) Керування прибутком та мінімізація ризиків.

Інвестиційний портфель — набір різних активів, які інвестор може придбати. Це наприклад, цінні папери, валюта, депозити та багато інших видів. Інвестор управляє цими активами, щоб досягти певних цілей.

При формуванні портфелю інвестори використовують диверсифікацію.

Диверсифікація — це коли інвестор розподіляє свій капітал між різними фінансовими активами, щоб в разі збитків в одному з них, інші могла покрити збитки.

Дивіденди — частина прибутку, яка виплачується власникам акцій в кінці кожного місяця.

Для виконання інвестиційно-фінансової діяльності створено багато різних додатків та інформаційних систем. Проте ідеєю цієї роботи є створення тренажеру з рекомендаційною системою для навчання індивідуальних інвесторів. За допомогою цієї системи будь-яка люди зможе навчитися правильно інвестувати свої власні кошти для отримання прибутку. Основною

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ідеєю є створення середовища максимально подібного до реального світового ринку (історичні дані), в якому інвестор має можливість з певною вибраною періодичністю впливати на свій портфель і дивитися можливі рекомендації по його зміні.

					ДП ІС-6127.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Опис предметного середовища

В даному предметному середовищі головним суб'єктом є інвестор. Який приймає рішення та управляє своїм портфелем, тобто активами. Активи – це об'єкти предметного середовища. За допомогою різних операцій над своїм портфелем в системі, інвестор може змінювати свій прибуток, мінімізувати ризики та робити передбачення.

Інвестор, визначившись з інвестиційною стратегією, повинен придбати той чи інший актив. Він прагне отримати прибуток з цього активу. Прибуток може бути різний: відсоток від депозиту, дивіденди від акцій компаній, прибуток від різниці в ціні акцій, валюти. Все це дає інвестору можливість при правильному підході отримати прибуток та збільшити свій капітал.

### 1.1.1 Опис процесу діяльності

Процес інвестування достатньо складний та ризикований. Не знаючи як правильно вкладати свій капітал та не дотримуючись певних правил та стратегій, можна втратити всі свої кошти.

Для того щоб повністю зрозуміти цей процес треба багато практики та знань. Проте практика в цьому на власних коштах може призвести до повної втрати своїх заощаджень.

### 1.1.2 Опис функціональної моделі

В даній системі єдиним актором буде користувач, який буде взаємодіяти з системою. Користувачу потрібно буде перейти на сторінку тренажеру, далі йому потрібно буде вибрати налаштування тренажеру та запустити тренажер.

В налаштування буде входити початковий капітал для вкладення. Кожен цикл користувач буде бачити свій баланс та всі його активи в портфелі з

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

певними характеристиками. Користувач зможе переглянути графіки зміни ціни активу та його характеристики.

Відповідно користувач зможе зробити зміни в свій портфель (продати/купити) та перейти до наступного кроку циклу.

Після кожної ітерації циклу користувач отримає новий результат, який сформувався після минулих змін.

В кінці кожного кроку циклу користувач побачить рекомендації щодо змін в портфелі для максимізації прибутку у майбутньому та мінімізації ризиків втрати капіталу.

Результатом остаточного кроку буде повна ціна портфелю та зароблені кошти за весь час моделювання.

Таблиця 1.1 – Функціональні вимоги

Функціональні вимоги	Пріоритет
1. Система надає користувачу можливість вибрати налаштування для процесу інвестування.	Високий
2. Система дозволяє переглянути дані інвестування. 2.1. Система надає користувачу можливість переглянути власний баланс та зміни відносно останнього кроку. 2.2. Система надає користувачу можливість переглянути свій портфель, та зміни по кожному активу.	Високий Високий Високий
3. Система дозволяє переглянути рекомендації щодо змін в портфелі.	Високий
4. Система дозволяє купити активи. 4.1. Система дозволяє відкрити депозит. 4.2. Система дозволяє купити акції компаній. 4.3. Система дозволяє купити валюту.	Високий Високий Високий Високий

5. Система дозволяє користувачу переглянути характеристики активу, що він хоче придбати.

Низький

## 1.2 Огляд наявних аналогів

Точного аналогу такої системи знайдено не було. Проте в той чи іншій мірі ця ідея реалізована в різних інформаційних системах. Наприклад, сайт investing.com при реєстрації надсилає на пошту рекомендації щодо поточної ситуації на ринку акцій. Він показує показник зміни ціни акцій за тиждень і робить припущення: ціна буде зростати і тому краще купувати чи ціна буде спадати тому краще продавати.

Існує багато різних брокерів, які дозволяють просто на демо-рахунках торгувати на біржі. Проте часто в таких системах іде розрахунок на людей, які інвестують в короткій перспективі.

Схожі системи починають впроваджуватись в банках різних країн світу. Які дозволяють переглядати власний портфель та отримувати дивіденди на власний рахунок. Гарними прикладами є «Тинькофф Инвестиции» або PayPal створили мобільний додаток для купівлі та продажу акцій. На жаль, в Україні такої можливості не існує. Для інвестування в Україні на біржі потрібні багато різних дозволів та договорів, що робить неможливим створення схожої системи на даний час.

Проте всі ці системи досить вузько направлені і не дають можливості в формі «гри» навчитися вкладати власні кошти та навчитися регулювати ризики, щоб кожен міг спробувати себе в цьому занятті.

## 1.3 Постановка задачі

### 1.3.1 Призначення розробки

Робота буде корисна всім хто бажає почати інвестувати та хоче на практичному прикладі навчитися розуміти графіки, реалізовувати

					ДП ІС-6127.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

інвестиційні стратегії, навчитися мінімізувати ризики та заробляти на власному капіталі.

В майбутньому ця система може стати першою в Україні системою, яка буде включати в собі всі можливості інвестицій. І кожен зможе вкласти кошти у те що захоче, не витрачаючи багато часу на пошук можливості інвестувати вільні кошти по різних банкам та системам.

### 1.3.2 Цілі та задачі розробки

Створити інформаційну систему, яка буде максимально простою для використання, щоб навіть людина, яка нічого не розуміє в фінансах могла навчилася робити певні операції та інвестувати.

Також важливо донести, що пасивний капітал не приносить нічого. З часом через інфляцію та інші можливі економічні ситуації можна втратити весь власний капітал. Наприклад, часто трапляється, що тримаючи всі кошти в одному банку чи в одній валюті, людина втрачає майже весь свій капітал, який збирала для купівлі квартири чи машини. Через це важливо кожному отримати базові навички в диверсифікації та інвестування в цілому.

В даній системі потрібно сформулювати такі економічні умови, які будуть максимально подібні до реальних речей в світі.

Зробити таку систему, щоб в майбутньому її можна було розвивати та впровадити в публічне використання. А також дати можливість розробляти інвестиційні стратегії на даних, які актуальні на даний момент часу в світі.

### Висновок до розділу

Процес інвестування складний та ризиковий. Часто для отримання якогось результату інвестору доводиться втрачати кошти через свої помилки. Тому метою даного проекту є створення легкого для розуміння та використання тренажеру, щоб кожен зміг спробувати себе у цій справі.

Основною ідеєю проекту є навчання людей процесу інвестування.

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11



## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Вхідні дані представлені у таблиці 2.1

Таблиця 2.1 – Вхідні дані

Назва даних	Опис
Історичні дані про ціни на акції та валюту.	Отримуються з сервісу Alpha Vantage та заносяться в базу.
Дані про налаштування процесу інвестування: початковий капітал.	Вводяться користувачем перед початком роботи системи.
Дані про сесію	Створюються під час початку роботи системи і змінюються під час її функціонування.

### 2.2 Вихідні дані

Вихідні дані представлені в таблиці 2.2.

Таблиця 2.2 – Вихідні дані

Назва даних	Опис
Рекомендації щодо зміни в портфелі	Різні рекомендації для користувача щодо покращення його портфелю.
Актуальні дані про сесію	Інформація про баланс портфелю, його склад та зміни з минулого кроку.
Дані про активи	Інформація та характеристики про активи, їх оцінка.

## 2.3 Опис структури бази даних

На початку проектування бази даних інформаційної системи були виділені сутності, що відображені у відповідних таблицях бази даних.

Схема бази даних наведена у графічних матеріалах.

- Розглянемо таблиці та зв'язки між ними. Таблиця Currencies (рисунок 2.2) – це таблиця, які містить назви валют та їх коди. Таблиця Stocks (рисунок 2.3) – це таблиця, які містить назви акцій та їх символи.


	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Code	nvarchar(MAX)	<input type="checkbox"/>
	CurrencyName	nvarchar(MAX)	<input type="checkbox"/>

Рисунок 2.2 –Таблиця Currencies


	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Symbol	nvarchar(MAX)	<input type="checkbox"/>
	CompanyName	nvarchar(MAX)	<input type="checkbox"/>

Рисунок 2.3 –Таблиця Stocks

Таблиця Sessions (рисунок 2.4) – це таблиця, яка містить інформацію про сесію. А саме її ідентифікатор та дата етапу інвестування.


	Column Name	Data Type	Allow Nulls
	SessionId	int	<input type="checkbox"/>
	CurrentDate	datetime2(7)	<input type="checkbox"/>

Рисунок 2.4 –Таблиця Sessions

Таблиця Portfolios (рисунок 2.5) відображає портфель користувача. Вона прив'язана до таблиці Sessions і показує поточний стан вашого портфелю, а саме вільні кошти та початковий капітал.

	Column Name	Data Type	Allow Nulls
PK	PortfolioId	int	<input type="checkbox"/>
	StartCapital	float	<input type="checkbox"/>
	FreeCapital	float	<input type="checkbox"/>
	SessionId	int	<input type="checkbox"/>

Рисунок 2.5 – Таблиця Portfolios

Таблиці CurrencyPrices (рисунок 2.6) та StockPrices (рисунок 2.7) містять історичні дані для валюти та акцій. Вони містять ціну, дату та прив'язані до таблиці Currencies та таблиці Stocks відповідно.

	Column Name	Data Type	Allow Nulls
PK	CurrencyPriceId	int	<input type="checkbox"/>
	Date	datetime2(7)	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
	CurrencyId	int	<input type="checkbox"/>

Рисунок 2.6 – Таблиця CurrencyPrices

	Column Name	Data Type	Allow Nulls
PK	StockPriceId	int	<input type="checkbox"/>
	Date	datetime2(7)	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
	StockId	int	<input type="checkbox"/>

Рисунок 2.7 – Таблиця StockPrices

Таблиця DepositPackages (рисунок 2.8) містить інформацію про відкриті депозити, а саме такі поля: скільки коштів біло вкладено, період у місяцях, час відкриття, відсотки річних та чи виводити прибуток на рахунок кожний місяць. Ця таблиця зв'язана з таблицею Portfolios.

	Column Name	Data Type	Allow Nulls
PK	DepositPackageId	int	<input type="checkbox"/>
	PortfolioId	int	<input type="checkbox"/>
	Cash	float	<input type="checkbox"/>
	Period	int	<input type="checkbox"/>
	[Open]	datetime2(7)	<input type="checkbox"/>
	Interest	float	<input type="checkbox"/>
	DeduceInterest	bit	<input type="checkbox"/>

Рисунок 2.8 – Таблиця DepositPackages

Таблиці CurrencyPackages (рисунок 2.9) та StocksPackages (рисунок 2.10) містять поточну інформацію про поточну ціну валюти та акцій та їх кількість в портфелі. Вони зв'язані з таблицею Portfolios та таблицями Currencies та таблиці Stocks відповідно.

	Column Name	Data Type	Allow Nulls
▶	CurrencyPackageld	int	<input type="checkbox"/>
	PortfolioId	int	<input type="checkbox"/>
	CurrencyId	int	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
	Cash	int	<input type="checkbox"/>

Рисунок 2.9 –Таблиця CurrencyPackages

	Column Name	Data Type	Allow Nulls
▶	StockPackageld	int	<input type="checkbox"/>
	PortfolioId	int	<input type="checkbox"/>
	StockId	int	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
	Count	int	<input type="checkbox"/>

Рисунок 2.10 –Таблиця StocksPackages

### Висновок до розділу

У розділі були описані вхідні та вихідні дані. Описана структура бази даних та пояснення до таблиці.

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Для реалізації програмного забезпечення були виділені наступні задачі:

- оптимізація портфелю користувача – максимізація прибутку від портфелю та мінімізація ризиків;
- обґрунтування знайдених покращень та створення рекомендацій для користувача.

Користувач має портфель, який містить різні види активів. Актив «акції» складається з кількості акцій та їх ціни. Актив «валюта» складається з кількості валюти та її ціни до долара. «Депозит» складається з суми вкладу та місячного доходу. Першою підзадачею даної системи є підбір таких значень кількості в активах, щоб максимізувати прибуток та другою підзадачею є підбір таких значень кількості, щоб мінімізувати ризики.

#### 3.2 Математична постановка задачі

Портфель складається з активів:

$$P = \sum_{i=1}^{n_c} C_i + \sum_{i=1}^{n_s} S_i + \sum_{i=1}^{n_d} D_i, \quad (1)$$

де  $C$  – валюта,  $S$  – акція компанії,  $D$  – відкритий депозит. А  $n_c$ ,  $n_s$  та  $n_d$  відповідно кількість типів валюти, акцій та депозитів.

Формула валюти:

$$C_i = x_i \cdot l_i, \quad i = \overline{1, n_c}, \quad (2)$$

де  $x$  – кількість купленої валюти, а  $l$  – ціна відносно долара.

Формула акції:

$$S_i = x_i \cdot l_i + m_i, \quad i = \overline{1, n_s}, \quad (3)$$

де  $x$  – кількість одиниць певної акції,  $l$  – ціна в доларах за одну,  $m$  – дивіденди.

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Формула депозиту:

$$D_i = \frac{x_i * u_i}{12}, \quad i = \overline{1, n_d}, \quad (4)$$

де  $x$  – кількість вкладених грошей.

Задачею є знаходження таких  $x_i$ , щоб максимізувати прибуток та мінімізувати ризики.

### 3.3 Обґрунтування методу розв'язання

Введемо функцію ризиків. Вона може бути будь-якою, бо для розв'язання задачі це не так важливо. В даній системі було вирішено використати таку функцію ризиків:

$$R(A) = \frac{\sum_i^t \omega_i \cdot U_i}{\sum_i^q \omega_i \cdot I_i} \quad (5)$$

Це зважена сума збитків( $U$ ) поділена на зважену суму доходів( $I$ ).  $\omega_i$  - вага доходу чи збитку.

$$\sum \omega_i = 1 \quad (6)$$

Ці ваги будуються по принципу: більш близькі події важливіші за старі події.

Функція  $R(A)$  може бути більше нуля, тоді кількість збитків більша за кількість доходів, тому робимо висновок, що актив досить ризиковий, а може бути менше нуля, що свідчить про достатньо високу імовірність отримати прибуток.

Тоді маємо функцію, яку потрібно мінімізувати.

$$\sum_{i=1}^{n_c} R_i \cdot x_i + \sum_{i=n_c}^{n_c+n_s} R_i \cdot x_i \rightarrow \min \quad (7)$$

Для неї маємо систему обмежень:

$$\begin{aligned}
 x_i &\geq 0.8 \cdot x'_i \cdot R_i \\
 x_i &\leq 1.2 \cdot x'_i \cdot R_i \\
 \sum_{i=1}^{n_c} l_i \cdot x_i &\leq 0.2 \cdot F \\
 \sum_{i=n_c}^{n_c+n_s} l_i \cdot x_i &\leq 0.2 \cdot F
 \end{aligned} \tag{8}$$

де  $x'_i$  - минула кількість активів, а  $F$  – це повний капітал. Останні два обмеження – це правило диверсифікації, яке не дає купити на всі кошти активи одного виду. Інші два обмеження – це обмеження на максимальну кількість куплених та проданих одиниць.

Друга підзадача зводиться до максимізації прибутку. Введемо поняття зважена середня ціна. Це ціна на актив помножена на ваги і поділена на суму ваг. Позначимо цю ціну, як  $G$ . Тоді маємо функцію, яку потрібно максимізувати.

$$\sum_{i=1}^{n_s} ((G_i - l_i + m_i) \cdot x_i) + \sum_{i=n_s}^{n_c+n_s} ((G_i - l_i) \cdot x_i) + \sum_{i=n_c+n_s}^{n_c+n_s+n_d} x_i \cdot u_i / 12 \rightarrow \max, \tag{9}$$

де  $(G_i - l_i)$  – різниця між зваженою середньою ціною та новою ціною.

Перша сума цієї функції – це для акцій, друга – для валюти та 3 – для депозиту.

Ця функція дає нам результат у вигляді місячного прибутку, який потрібно максимізувати.

Обмеження використаємо такі самі, що і в минулій функції. Крім обмежень на депозит, тут просто будемо вважати, щоб вклад був менше за 20 відсотків від повної вартості капіталу.

Отримані значення змінних треба привести до рекомендацій за такими правилами: якщо невідома змінна зменшилась треба повідомити користувача

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

про те, що актив не прибутковий і краще продати частину цього активу, якщо ж збільшилась – порадити придбати ще одиниць цього активу.

### 3.4 Опис методів розв'язання

Для знаходження розв'язку двох цих функцій будемо використовувати симплекс метод [1].

$$c^T x \rightarrow \max (\min) \quad (10)$$

$$Ax \leq b \quad (11)$$

В нашому випадку є функції, які ми оптимізуємо, обмеження також є. Залишилося розв'язати задачі та обґрунтувати знайдені результати.

Виконаємо вимоги стандартної форми запису задачі лінійного програмування. Для цього перетворимо всі обмеження в рівності з невід'ємною правою частиною і забезпечимо невід'ємність усіх змінних.

Побудуємо і заповнимо початкову симплекс таблицю, яка є зручним інструментом для подання канонічної форми лінійної задачі. Для заповнення початкової симплекс таблиці необхідно переписати цільову функцію F у вигляді, аналогічному до системи обмежень (рисунок 3.1).

Проведемо перевірку на оптимальність – якщо всі коефіцієнти в рядку F є невід'ємними, то отриманий розв'язок є оптимальним, але якщо хоча б один коефіцієнт є від'ємний, то необхідно продовжити симплекс ітерацію, тобто заповнити наступну симплекс таблицю.

Оберемо стовпчик, в якому міститься найбільший за модулем від'ємний коефіцієнт в рядку F (ведучий стовпчик).

Оберемо ведучий рядок та ведучий елемент  $a_{rs}$ . Для цього скористаємось скористатись тестом мінімального відношення (MRT) – запишемо у відповідному рядку відношення змінної зі стовпця "план" до змінної з ведучого стовпця і визначимо мінімальне з цих відношень. Рядок з мінімальним значенням відношення буде ведучим рядком.

					ДП ІС-6127.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		



Модифікуємо симплекс таблицю по відношенню до ведучого елемента

$a_{rs}$ :

- для ведучого рядка:

$$\widehat{a}_{rj} = \frac{a_{rj}}{a_{rs}^*}; \quad (12)$$

- для ведучого стовпця:

$$\widehat{a}_{rs} = 1; \widehat{a}_{is} = 0; j = \overline{1, m}; i \neq r; \quad (13)$$

- для решти елементів:

$$\widehat{a}_{ij} = \frac{a_{ij} \cdot a_{rs}^* - a_{rj} \cdot a_{is}}{a_{rs}^*} \quad (14)$$

Повертаємося до перевірки на оптимальність.

Базис	План	$x_1$	$x_2$	...	$x_s$	...	$x_n$	$x_{n+1}$	...	$x_{n+m}$	MRT
F	0	$-c_1$	$-c_2$	...	$-c_s$	...	$-c_n$	0	...	0	
$x_{n+1}$	$b_1$	$a_{11}$	$a_{12}$	...	$a_{1s}$	...	$a_{1n}$	1	...	0	$\frac{b_1}{a_{1s}}$
...											
$x_{n+r}$	$b_r$	$a_{r1}$	$a_{r2}$	...	$a_{rs}$	...	$a_{rn}$	0	...	0	$\frac{b_r}{a_{rs}}$
$x_{n+m}$	$b_m$	$a_{m1}$	$a_{m2}$	...	$a_{ms}$	...	$a_{mn}$	0	...	1	$\frac{b_m}{a_{ms}}$

Рисунок 3.1 – Симплекс таблиця

Кінцевим результатом роботи є базисні змінні та цільова функція. Ці дані потребують обґрунтування та подальшого використання.

### Висновок до розділу

У розділі математичного забезпечення були оглянуті математичні задачі дипломного проекту. Розглянуто роботу симплекс методу та описано, як використати його в даній задачі.

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Програмне забезпечення складається з двох рівних частин, а саме з клієнтської та серверної.

В основному для розробки серверної частини було використано ASP.NET Core - це фреймворк з відкритим вихідним кодом, розроблений Microsoft і її спільнотою. Він є крос-платформений, а це означає що програми написані на цьому фреймворку можна запускати на будь-якій операційній системі, що дає багато можливостей на його використання.

Функціонал цього фреймворку дозволяє створити веб-застосунок, який буде незалежний від клієнтської частини, бо запити до серверу можна буде робити по http. А це дасть нам повну незалежність сервер від клієнта.

Для використання бази даних було використано **Entity Framework Core** — це найлегший спосіб доступу до бази в C#. EF Core надає можливість використовувати правило “Code First”, що дозволяє в коді створювати класи та модель системи, а потім на основі цього будується база. Розробнику не потрібно спочатку проектувати базу, а достатньо написати код для її створення. Також ця технологія дозволяє робити запити в базу, не використовуючи великі SQL-запити, що зменшує кількість коду та полегшує розробку.

Для того, щоб отримати історичні дані було використано **Alpha Vantage API**. Це REST API надає багато різних даних та характеристик, у тому числі історичних, по валютам, акціям та іншим активам.

Для розробки клієнтської частини було використано Angular, HTML, CSS, Bootstrap.

**Angular** — TypeScript front-end фреймворк з відкритим кодом. Він розробляється у компанії Google. Через те, що він дозволяє писати на TypeScript, розробка полегшується.

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

**HTML** — мова розмітки документів. Використовується як розмітка для сторінок веб-застосунку.

**CSS** — це спеціальна мова стилів, яка дозволяє додавати певні стилі для сторінок.

**Bootstrap** — це набір інструментів, які мають відкритий код, вони використовуються для розробки веб-сайтів та веб-додатків. Ці інструменти містять шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Шаблон сайту та деякі компоненти клієнта були взяті з шаблону **Argon Dashboard Angular**, яка має відкритий вихідний код та поставляється під ліцензією MIT. Ліцензія міститься в коді роботи. [3]

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Для роботи серверу необхідні наступні параметри системи:

- процесор с частотою 2.4 ГГц або більш потужний;
- ОЗУ об'ємом 4 ГБ;
- мінімум 10 Гб доступного простору на жорстким диску;
- жорсткий диск с частотою обертання 5 400 об/хв або більше.

Також необхідне встановлене наступне програмне забезпечення:

- Ubuntu 18.04 або Windows;
- MSSQL;
- ASP.NET Core Runtime 2.2.0.

Вимоги до клієнту – це наявність одного з веб-браузерів, який підтримує Angular (наприклад, Google Chrome 83+).

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

### 4.3 Архітектура програмного забезпечення

#### 4.3.1 Діаграма класів

На діаграмі класів наведено модель системи, яка відповідає за оновлення даних та формування даних для використання у рекомендаціях. Тут є 3 класи CurrencyDataUpdater, DepositDataUpdater та StockDataUpdater, які оновлюють всю інформацію. Класи Currency, Stock та Deposit мають методи та характеристики, які треба оновлювати кожний крок.

Діаграма класів знаходиться у графічних матеріалах.

#### 4.3.2 Діаграма послідовності

На діаграмі послідовності показано основну роботу програми. Спочатку користувач за допомогою клієнта робить запит для того, щоб розпочати сесію. Клієнтська частина передає HTTP POST запит до серверу. Сервер створює запис в базі та вертає ідентифікатор сесії. Далі користувач попадає на сторінку сесії. Тут він може купити активи та за допомогою POST запитів додати необхідні активи в портфель, при чому сервер додає ці записи в базу.

Останнім етапом є перехід до наступного кроку. Клієнт робить POST запит, сервер дістає всі дані про портфель сесії з бази, потім дістає нові дані та формує пакет даних, який потім використовується для знаходження рекомендацій. Останнім кроком сервер оновлює базу новими даними та повертає отримані результати клієнту.

Діаграма послідовності знаходиться у графічних матеріалах.

#### 4.3.3 Діаграма розгортання

На діаграмі розгортання зображені ключові компоненти сервера:

- Flask Application – серверна частина проекту;
- MSSQL Database – база даних.

А також компоненти клієнта:

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

– Web browser – браузер користувача.

Діаграма розгортання наведена у графічних матеріалах.

#### 4.3.4 Специфікація функцій

Назва	Примітка
<b>Клас DataProviderService</b> – сервіс що відповідає за інформацію про валюту та акції	
Task<List<Currency>> GetCurrencies()	Дістає з бази назву валюти та її код
Task<List<Stock>> GetStocks()	Дістає з бази назву акції та її символ
<b>Клас StocksService</b> – сервіс що відповідає за операції над акціями в портфелі	
Task Buy(int portfolioId, string symbol, int count)	Додає акції в портфель
Task Sell(int portfolioId, string symbol, int count)	Видаляє акції з портфеля
<b>Клас CurrenciesService</b> – сервіс що відповідає за операції над валютою в портфелі	
Task Buy(int portfolioId, string code, double cash)	Додає валюту в портфель
Task Sell(int portfolioId, string code, double cash)	Видаляє валюту з портфеля
<b>Клас DepositService</b> – сервіс що відповідає за операції над депозитами в портфелі	
Task Open(int portfolioId, double cash, double interest, int period, Boolean deduceInterest)	Відкриває депозит
<b>Клас SessionService</b> – сервіс що відповідає за сесію	
Task<string> Create(double startCapital)	Створює сесію
Task<SessionSnapshot > NextStep(int sessionId)	Оновлює всі дані, створює рекомендації

**Висновок до розділу**

У розділі біли розглянуті засоби розробки системи, вимоги до технічного забезпечення. А також детально розглянуто модель системи у вигляді діаграми класів та за допомогою специфікацій функцій. Та показано послідовність дій та інфраструктуру системи.

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

На головній сторінці знаходиться форма для налаштування сесії (рисунок 5.1). Як тільки нажимається кнопка «РОЗПОЧАТИ» головна сторінка змінюється на сторінку сесії, на якій зображується основна інформація по сесії, а також рекомендації та портфель користувача (рисунок 5.2).

Рисунок 5.1 – Головна сторінка, поки сесія не розпочата

НАЗВА АКТИВУ	ПОВНА ЦІНА АКТИВУ	ПРИЗБОТ ЗА МІСЯЦЬ	ВІДСОТК ЗМІНИ ЦІНИ АКТИВУ
Deposit 5%	4000\$	12\$	0%
Facebook stocks	2000\$	12\$	-3%
Apple stocks	3000\$	16\$	5%
Euro currency	300\$	0\$	2%

Рисунок 5.2 – Сторінка сесії

На сторінці сесії можливо додати активи натиснувши на кнопку «Додати новий актив». Тоді відкривається сторінка з можливими активами (рисунок 5.3).

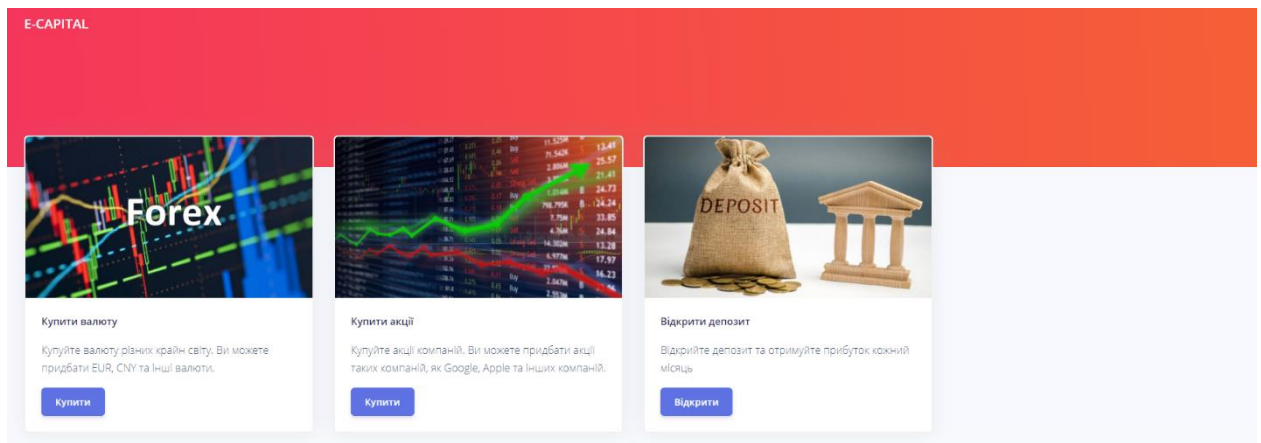


Рисунок 5.3 – Сторінка вибору активу

На сторінці вибору активу (рисунок 5.3) можна вибрати один з трьох можливих активів: депозит, акції або валюта. Після вибору необхідного активу треба натиснути відповідну кнопку.

На сторінці відкриття депозиту (рисунок 5.4) знаходяться всі необхідні налаштування, щоб відкрити депозит.

The screenshot shows the E-CAPITAL website with the 'Депозит' (Deposit) page. It includes a definition of a deposit and configuration options:

**Депозит**  
Депозит - це зберігання ваших заощаджень у банку під відсоток. За визначенням, депозит або вклад - це угода, де одна сторона (банк), яка приймає від другої сторони (вкладника) або для неї певні кошти (вклад), зобов'язується виплачувати вкладникові всю суму вклату та виплатити відсотки на неї або дохід в іншій формі на умовах та в порядку, встановлених договором.

**Налаштування**

Скільки бажаєте покласти?  
1035.00 (US \$)

На скільки місяців відкривати?  
5

Річних, %  
3

Що робити з прибутком?  
☐ Виплачувати на рахунок  
☒ Додавати до суми депозиту

Ви отримаєте прибутку після завершення строку вкладення:  
13 (US \$)

Зуважте, що депозит не можна буде закрити раніше кінця його строку. Тому кошти, які ви зараз вкладаєте будуть недоступні для використання наступні 5 місяці!

Відкрити

Рисунок 5.4 – Сторінка відкриття депозиту



На сторінці купівлі акцій (рисунок 5.5) можна купити або продати акції компаній, для цього необхідно зробити відповідні дії на правій та лівій картці – «Додати в кошик», «Видалити».

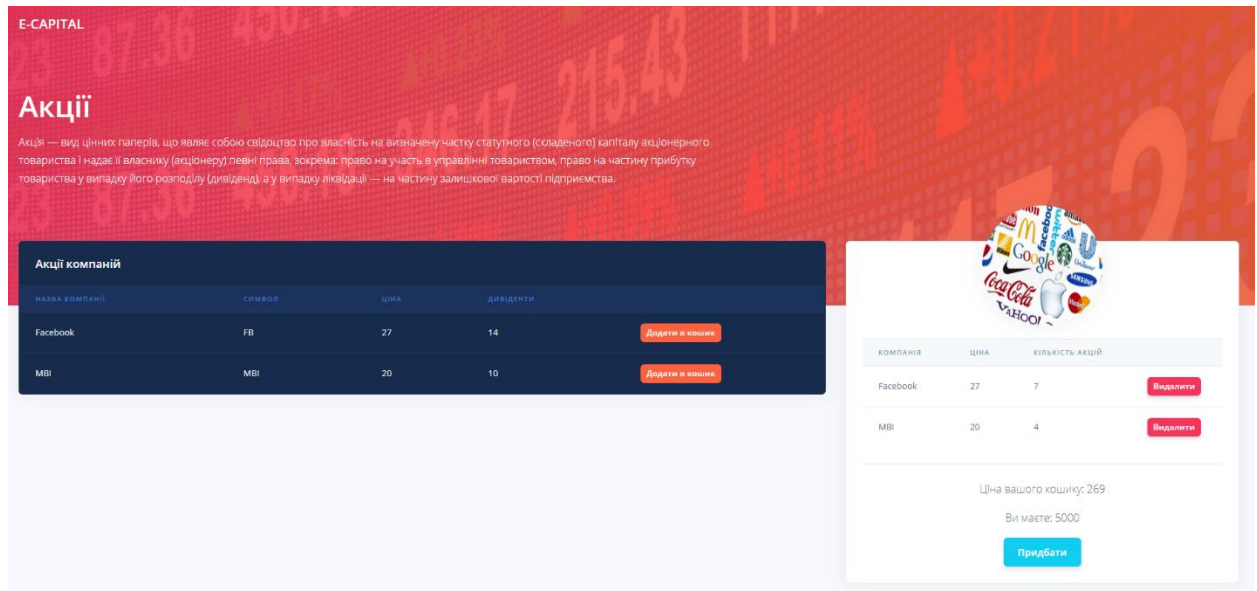


Рисунок 5.5 – Сторінка купівлі/продажі акцій

На сторінці купівлі валюти (рисунок 5.6) можна купити або продати валюту країн світу, для цього необхідно зробити відповідні дії на правій та лівій картці – «Додати в кошик», «Видалити» та записати кількість валюти, яку треба купити/продати.

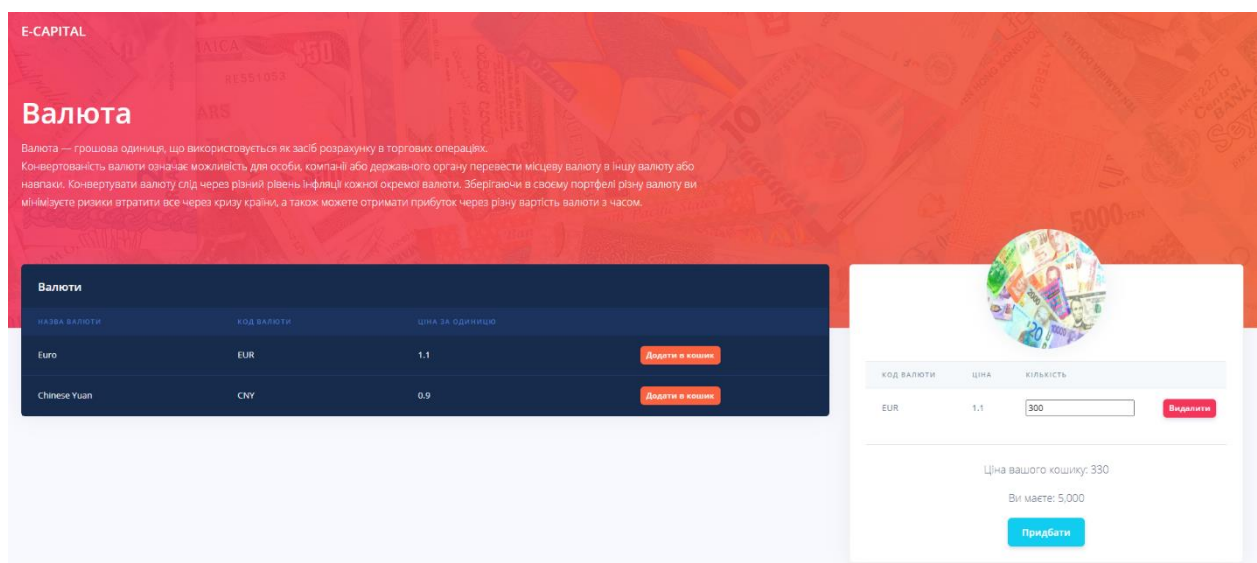


Рисунок 5.6 – Сторінка купівлі/продажі валюти

## 5.2 Випробування програмного продукту

### 5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій інформаційної система з підтримки інвестиційно-фінансової діяльності вимогам технічного завдання.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

Під час тестування була перевірена функціональна складова системи. Нижче наведені результати випробувань у вигляді таблиць.

**Таблиця 5.1** – Початок сесії

Мета тесту	Перевірка початку сесії
Початковий стан моделі	Початкова сторінка веб-застосунку
Схема проведення тесту	Ввести початковий капітал і натиснути «РОЗПОЧАТИ».
Очікуваний результат	Екран сесії
Результат роботи	Екран сесії

Таблиця 5.2 – Додати актив

Мета тесту	Перевірка форми з додаванням активу
Початковий стан моделі	Сторінка сесії
Схема проведення тесту	Натиснути кнопку «Додати новий актив» на картці «Ваші активи»
Очікуваний результат	Екран вибору активу
Результат роботи	Екран вибору активу

Таблиця 5.3 – Відкрити депозит

Мета тесту	Перевірка можливості відкриття депозиту
Початковий стан моделі	Сторінка відкриття депозиту
Схема проведення тесту	Ввести кількість грошей, які потрібно покласти, вибрати кількість місяців, вибрати чи додавати відсотки до рахунку і натиснути кнопку «Відкрити»
Очікуваний результат	Закриття екрану з депозитом і повернення на екран сесії
Результат роботи	Закриття екрану з депозитом і повернення на екран сесії, депозит з'явився в списку наявних активів

Таблиця 5.4 – Купити акції/валюту

Мета тесту	Перевірка можливості купівлі акцій/валюти
Початковий стан моделі	Сторінка акцій/валюти
Схема проведення тесту	На картці «Акції компаній»/«Валюти» біля необхідної акції/валюти натиснути кнопку «Додати в кошик». Натиснути кнопку «Придбати»/ввести кількість валюти на натиснути кнопку «Придбати».
Очікуваний результат	Закриття екрану з акціями/валютою і повернення на екран сесії
Результат роботи	Закриття екрану з акціями/валютою і повернення на екран сесії, відповідні активи з'явилися в списку наявних активів

Таблиця 5.5 – Перехід до наступного кроку

Мета тесту	Перевірка переходу до наступного кроку
Початковий стан моделі	Сторінка сесії
Схема проведення тесту	Натиснути кнопку «Наступний крок»
Очікуваний результат	Сторінка завантаження, а потім сторінка сесії
Результат роботи	Сторінка завантаження, а потім сторінка сесії. На новій сторінці змінились дані, додалися рекомендації.

Таблиця 5.6 – Перегляд даних про акцію/валюту

Мета тесту	Перевірка сторінки з інформацією про акцію/валюту
Початковий стан моделі	Сторінка акцій/валюти
Схема проведення тесту	На картці «Акції компаній»/«Валюти» натиснути на необхідну акцію/валюту
Очікуваний результат	Сторінка інформації з графіками та даними
Результат роботи	Сторінка інформації з графіками та даними

**Висновок до розділу**

У розділі було розглянуто інструкцію користувача, ознайомлення з можливостями системи. Було успішно виконане тестування головних функцій системи.

**ЗАГАЛЬНІ ВИСНОВКИ**

У сучасному світі інвестування має досить велике значення, адже кожен прагне зберегти свої гроші та прибільшити їх. Але процес інвестування складний та ризиковий. За неправильного інвестування можна втратити всі кошти. Тому ідеєю цієї роботи є створення тренажеру, який дозволить спробувати себе у інвестуванні і навчитися цьому процесу.

В роботі розглянуто математичну постановку задачі, а саме математику для рекомендацій. Побудовано вимоги до системи та розроблено тести для неї.

Система була створена достатньо гнучкою для того, щоб у майбутньому її можна було розвивати та удосконалювати.

Ця робота буде корисна для ознайомлення будь-кому, хто захоче спробувати себе у інвестуванні на реальних прикладах із життя.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Хемди А. Таха Глава 3. Симплекс-метод. *Введение в исследование операций*. 7-е изд. М.: "Вильямс", 2007.
2. Томас Х. Кормен и др. Глава 29. Линейное программирование. *Алгоритмы: построение и анализ*. 2-е изд. М.: «Вильямс», 2006. С. 1296. ISBN 5-8459-0857-4.
3. Argon Dashboard Angular [Электронный ресурс] – Режим доступа до ресурсу:  
<https://github.com/creativetimofficial/argon-dashboard-angular.git>.

## Додаток А

**Тексти програмного коду**Інформаційна система з підтримки інвестиційно-фінансової діяльності

(Найменування програми (документа))

DVD-R

(Вид носія даних)

8 арк, 244 Кб

(Обсяг програми (документа) , арк.,) Кб)

КИЇВ – 2020 РОКУ

					ДП ІС-6127.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

## Лістинг основної частини програми

```
public abstract class Asset
{
    public abstract AssetType Type { get; }
    public decimal Risk { get; protected set; }
    public decimal WeightedAverage { get; protected set; }
    public decimal LastUpdateIncome { get; protected set; }
}

public enum AssetType
{
    Deposit = 0,
    Currency = 1,
    Stock = 2
}

public interface IDataUpdater
{
    Task Update(Asset asset);
}

public class Currency : Asset
{
    public override AssetType Type => AssetType.Currency;

    public string FromSymbol { get; }
    public string ToSymbol => "USD";

    public decimal Price { get; private set; }
    public decimal Value { get; private set; }
    public decimal Sum { get; private set; }

    public DateTime LastUpdatedTime { get; private set; }

    public void UpdateTime()
    {
        LastUpdatedTime = LastUpdatedTime.AddMonths(1);
    }

    public void Add(decimal value)
    {
        Value += value;
    }

    public void ChangePrice(decimal newPrice)
    {
        Price = newPrice;
    }

    public void ChangeSum()
    {
        Sum = Value * Price;
    }
}

public class CurrencyDataUpdater : IDataUpdater
```



```

private IMonthlyCurrencyDataService _monthlyCurrencyDataService;

public CurrencyDataUpdater()
{
    _monthlyCurrencyDataService = new MonthlyCurrencyDataService(new ApiClient());
}

public async Task Update(Asset asset)
{
    var currency = asset as Model.Currency;

    var data = await _monthlyCurrencyDataService.GetData(currency.FromSymbol, currency.ToSymbol,
        CancellationToken.None);

    currency.UpdateTime();

    var newData = data.MonthlyTimeSeries.FirstOrDefault(o => o.Key == currency.LastUpdatedTime);
    currency.ChangePrice(decimal.Parse(newData.Value.Close));
    currency.ChangeSum();
}

public class Deposit : Asset
{
    public decimal Value { get; }
    public decimal Income { get; private set; }
    public decimal Interest { get; }
    public bool DeduceInterest { get; }
    public int MonthToHold { get; }
    public int MonthCompleted { get; private set; }

    public bool MustBeClosed => MonthToHold == MonthCompleted;

    public override AssetType Type => AssetType.Deposit;

    public Deposit(decimal value, decimal interest, bool deduceInterest, int monthToHold)
    {
        Interest = interest;
        Value = value;
        DeduceInterest = deduceInterest;
        MonthToHold = monthToHold;
    }

    internal void AddIncome(decimal value)
    {
        MonthCompleted++;
        Income += value;
        if (DeduceInterest)
        {
            LastUpdateIncome = value;
        }
    }
}

public class DepositDataUpdater : IDataUpdater

```

```

        private readonly IDepositOperations _depositOperations;

        public DepositDataUpdater()
        {
            _depositOperations = null;
        }

        public Task Update(Asset asset)
        {
            var deposit = asset as Model.Deposit;
            var income = _depositOperations.CalculateMonthIncome(deposit.Value, deposit.MonthToHold,
deposit.Interest);
            deposit.AddIncome(income);
            return Task.CompletedTask;
        }
    }

    public class Stocks : Asset
    {
        public override AssetType Type => AssetType.Stock;

        public int Count { get; private set; }
        public string Symbol { get; private set; }
        public decimal Price { get; private set; }
        public decimal MonthDividends { get; }
        public decimal Sum { get; private set; }

        public DateTime LastUpdateTime { get; private set; }

        public Stocks(string symbol, int count, decimal price)
        {
            LastUpdateTime = new DateTime(2010, 1, 1);
            Symbol = symbol;
            Price = price;
            Count = count;
            MonthDividends = LastUpdateIncome = new Random().Next(5, 20);
        }

        public void UpdateTime()
        {
            LastUpdateTime = LastUpdateTime.AddMonths(1);
        }

        public void AddStocks(int count)
        {
            Count += count;
        }

        public void ChangePrice(decimal newPrice)
        {
            Price = newPrice;
        }

        public void ChangeSum()
        {
            Sum = Count * Price;
        }
    }

```

```

public class StockDataUpdater : IDataUpdater
{
    private MonthlyStockDataService _monthlyStockDataService;

    public StockDataUpdater()
    {
        _monthlyStockDataService = new MonthlyStockDataService(new ApiClient());
    }

    public async Task Update(Asset asset)
    {
        var stock = asset as Model.Stocks;
        var data = await _monthlyStockDataService.GetData(stock.Symbol, CancellationToken.None);

        stock.UpdateTime();

        var newData = data.MonthlyTimeSeries.FirstOrDefault(o => o.Key == stock.LastUpdatedTime);
        stock.ChangePrice(decimal.Parse(newData.Value.Close));
        stock.ChangeSum();
    }
}

public class MonthlyCurrencyData
{
    [JsonProperty("Meta Data")]
    public MetaData MetaData { get; set; }

    [JsonProperty("Time Series FX (Monthly)")]
    public Dictionary<DateTime, MonthlyTimeSery> MonthlyTimeSeries { get; set; }
}

public class MetaData
{
    [JsonProperty("1. Information")]
    public string Information { get; set; }

    [JsonProperty("2. From Symbol")]
    public string FromSymbol { get; set; }

    [JsonProperty("3. To Symbol")]
    public string ToSymbol { get; set; }

    [JsonProperty("4. Last Refreshed")]
    public DateTimeOffset LastRefreshed { get; set; }

    [JsonProperty("5. Time Zone")]
    public string TimeZone { get; set; }
}

public class MonthlyTimeSery
{
    [JsonProperty("1. open")]
    public string Open { get; set; }

    [JsonProperty("2. high")]
    public string High { get; set; }

    [JsonProperty("3. low")]

```

```
[JsonProperty("4. close")]
```

```
    public string Close { get; set; }
}
```

```
public interface IMonthlyCurrencyDataService
```

```
{
    Task<MonthlyCurrencyData> GetData(string fromSymbol, string toSymbol, CancellationToken
cancellationToken);
}
```

```
public class MonthlyCurrencyDataService : IMonthlyCurrencyDataService
```

```
{
    private readonly IApiClient _apiClient;
    private const string PARAM_SEPARATOR = "&";
    private const string FUNCTION = "function=FX_MONTHLY";

```

```
    public MonthlyCurrencyDataService(IApiClient apiClient)
    {
        _apiClient = apiClient;
    }

```

```
    public Task<MonthlyCurrencyData> GetData(string fromSymbol, string toSymbol, CancellationToken
cancellationToken)
    {
        return _apiClient.Get<MonthlyCurrencyData>(CreateParams(fromSymbol, toSymbol), cancellationToken);
    }

```

```
    private string CreateParams(string fromSymbol, string toSymbol)
    {
        return
        $" {FUNCTION} {PARAM_SEPARATOR} from_symbol={fromSymbol} {PARAM_SEPARATOR} to_symbol={to
Symbol} ";
    }
}
```

```
public interface IApiClient
```

```
{
    Task<TResponse> Get<TResponse>(string params, CancellationToken cancellationToken);
}
```

```
public class ApiClient : IApiClient
```

```
{
    private const string PARAM_SEPARATOR = "?";
    private const string ACCESS_TOKEN = "&apikey=";

    public async Task<TResponse> Get<TResponse>(string params, CancellationToken cancellationToken)
    {
        using (var client = new HttpClient() { BaseAddress = new Uri("https://www.alphavantage.co/query") })
        {
            var response = await client.GetAsync(UriFormatter(params));
            response.EnsureSuccessStatusCode();
            var content = await response.Content.ReadAsStringAsync();
            return JsonConvert.DeserializeObject<TResponse>(content);
        }
    }
}
```

```
    private string UriFormatter(string params)
```

					ДП ІС-6127.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

return $" {PARAM_SEPARATOR} {parametrs} {ACCESS_TOKEN}";

    }
}

public class AnalyzerService : IAnalyzerService
{
    public Task<List<string>> GetRecommendation(decimal total, List<Deposit> deposits, List<Stocks> stocks,
List<Currency> currencies)
    {
        List<string> recommendations = new List<string>();
        recommendations.AddRange(GetRiskMinimizationRecommendations(total, stocks, currencies));
        recommendations.AddRange(GetIncomeMaximizationRecommendations(total, stocks, currencies,
deposits));

        return Task.FromResult(recommendations);
    }

    private List<string> GetRiskMinimizationRecommendations(decimal total, List<Stocks> stocks,
List<Currency> currencies)
    {
        var variablesCount = stocks.Count + currencies.Count;

        var constraints = new List<Constraint>();
        constraints.AddRange(GetStockConstraints(total, variablesCount, stocks));
        constraints.AddRange(GetCurrencyConstraints(total, variablesCount, stocks.Count * 2 + 1, currencies));

        var function = GetMinimizeFunction(stocks, currencies);

        var simplexAlgorithm = new SimplexAlgorithm(function, constraints.ToArray());
        var result = simplexAlgorithm.GetResult();

        return new List<string>();
    }

    private List<string> GetIncomeMaximizationRecommendations(decimal total, List<Stocks> stocks,
List<Currency> currencies, List<Deposit> deposits)
    {
        var variablesCount = stocks.Count + currencies.Count;

        var constraints = new List<Constraint>();
        constraints.AddRange(GetStockConstraints(total, variablesCount, stocks));
        constraints.AddRange(GetCurrencyConstraints(total, variablesCount, stocks.Count * 2 + 1, currencies));
        constraints.AddRange(GetDepositConstraints(total, variablesCount,
stocks.Count * 2 + currencies.Count * 2 + 2, deposits));

        var function = GetMaximizeFunction(stocks, currencies, deposits);

        var simplexAlgorithm = new SimplexAlgorithm(function, constraints.ToArray());
        var result = simplexAlgorithm.GetResult();

        return new List<string>();
    }
}

```

					ДП ІС-6127.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private Function GetMaximizeFunction(List<Stocks> stocks, List<Currency> currencies, List<Deposit> deposits)
{
    double[] variables = new double[stocks.Count + currencies.Count + deposits.Count];
    var index = 0;
    foreach (var stock in stocks)
    {
        variables[index] = (double)(stock.WeightedAverage - stock.Price + stock.MonthDividends);
        index++;
    }
    foreach (var currency in currencies)
    {
        variables[index] = (double)(currency.WeightedAverage - currency.Price);
        index++;
    }
    foreach(var deposit in deposits)
    {
        variables[index] = (double)(deposit.Interest / 12);
        index++;
    }

    return new Function(variables, 0, false);
}

private Function GetMinimizeFunction(List<Stocks> stocks, List<Currency> currencies)
{
    double[] variables = new double[stocks.Count + currencies.Count];
    var index = 0;
    foreach (var stock in stocks)
    {
        variables[index] = (double)stock.Risk;
        index++;
    }
    foreach (var currency in currencies)
    {
        variables[index] = (double)currency.Risk;
        index++;
    }

    return new Function(variables, 0, false);
}

private Constraint[] GetStockConstraints(decimal total, int variablesCount, List<Stocks> stocks)
{
    var constraints = new List<Constraint>();
    var index = 0;
    foreach (var stock in stocks)
    {
        var variables = new double[variablesCount];
        variables[index] = 1;
        constraints.Add(new Constraint(variables, 0.8 * stock.Count, ">="));
        constraints.Add(new Constraint(variables, 1.2 * stocks.Count, "<="));
        index++;
    }

    var divVariables = new double[variablesCount];

```

```

index = 0;

    foreach (var stock in stocks)
    {
        divVariables[index] = (double)stock.Price;
        index++;
    }
    constraints.Add(new Constraint(divVariables, 0.4 * (double)total, "<="));

    return constraints.ToArray();
}

private Constraint[] GetCurrencyConstraints(decimal total, int variablesCount, int startIndex, List<Currency>
currencies)
{
    var constraints = new List<Constraint>();
    var index = startIndex;
    foreach (var currency in currencies)
    {
        var variables = new double[variablesCount];
        variables[index] = 1;
        constraints.Add(new Constraint(variables, 0.8 * (double)currency.Value, ">="));
        constraints.Add(new Constraint(variables, 1.2 * (double)currency.Value, "<="));
        index++;
    }

    var divVariables = new double[variablesCount];
    index = startIndex;
    foreach (var currency in currencies)
    {
        divVariables[index] = (double)currency.Price;
        index++;
    }
    constraints.Add(new Constraint(divVariables, 0.3 * (double)total, "<="));

    return constraints.ToArray();
}

private Constraint[] GetDepositConstraints(decimal total, int variablesCount, int startIndex, List<Deposit>
deposits)
{
    var constraints = new List<Constraint>();
    var index = startIndex;
    var divVariables = new double[variablesCount];
    foreach (var deposit in deposits)
    {
        divVariables[index] = 1;
        index++;
    }
    constraints.Add(new Constraint(divVariables, 0.3 * (double)total, "<="));

    return constraints.ToArray();
}

}

public interface IAnalyzerService
{
    Task<List<string>> GetRecommendation(decimal total,
        List<Deposit> deposits, List<Stocks> stocks, List<Currency> currencies);
}

```

					ДП ІС-6127.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації та управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_ О.В. Ковтунець  
(підпис) (ініціали, прізвище)

“13” квітня 2020 р.

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри**

\_\_\_\_\_ О.А. Павлов  
(підпис) (ініціали, прізвище)

“14” квітня 2020 р.

«Інформаційна система з підтримки інвестиційно-фінансової  
діяльності»

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр ДП ІС-6127.01.000 ТЗ

на 10 сторінках

Київ – 2020 року



## ЗМІСТ

<b>1</b>	<b>ЗАГАЛЬНІ ПОЛОЖЕННЯ .....</b>	<b>3</b>
1.1	Повне найменування системи та її умовне позначення.....	3
1.2	Найменування організації-замовника .....	3
1.3	Перелік документів, на підставі яких створюється система.....	3
1.4	Планові терміни початку і закінчення роботи зі створення системи .....	4
<b>2</b>	<b>ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ СИСТЕМИ.....</b>	<b>5</b>
2.1	Призначення системи.....	5
2.2	Цілі створення системи.....	5
<b>3</b>	<b>ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....</b>	<b>6</b>
<b>4</b>	<b>ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>6</b>
4.1	Вимоги до функціональних характеристик.....	7
4.2	Вимоги до надійності.....	7
4.3	Вимоги до складу і параметрів технічних засобів.....	8
<b>5</b>	<b>СТАДІЇ І ЕТАПИ РОЗРОБКИ .....</b>	<b>9</b>
<b>6</b>	<b>ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ .....</b>	<b>10</b>
6.1	Види випробувань.....	10

					<b>ДП ІС-6127.01.000 ТЗ</b>			
<b>Зм.</b>	<b>Арк.</b>	<b>Прізвище</b>	<b>Підпис</b>	<b>Дата</b>	Інформаційна система з підтримки інвестиційно-фінансової діяльності	<b>Лім.</b>	<b>Лист</b>	<b>Листів</b>
Розробив		Ходарченко А.А.					2	10
Перевірив.		Ковтунець О.В				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61		
Н. кон.		Телишева Т.О.						
Затв.		Павлов О.А.						

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Повне найменування системи та її умовне позначення

**Повна назва системи:** Інформаційна система з підтримки інвестиційно-фінансової діяльності.

### 1.2 Найменування організації-замовника

Генеральним замовником проекту являється кафедра Автоматизованих систем обробки інформації та управління НТУУ "КПІ". Представником замовника є Ковтунець Олесь Володимирович.

Розробником системи є студент групи ІС-61 факультету інформатики та обчислювальної техніки НТУУ «КПІ ім. Ігоря Сікорського» Ходарченко Андрій.

### 1.3 Перелік документів, на підставі яких створюється система

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

					ДП ІС-6127.01.000 ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку роботи над створенням інформаційної системи з підтримки інвестиційно-фінансової діяльності – 10 лютого 2020 рік.

Плановий термін по закінченню роботи над створенням системи інформаційної системи з підтримки інвестиційно-фінансової діяльності – не пізніше 1 червня 2020 року.

					ДП ІС-6127.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРИЗНАЧЕННЯ І МЕТА СТВОРЕННЯ СИСТЕМИ

### 2.1 Призначення системи

Система буде надавати можливість користувачу спробувати себе у інвестиційній діяльності.

### 2.2 Цілі створення системи

Створити інформаційну систему, яка буде максимально простою для використання, щоб навіть людина, яка нічого не розуміє в фінансах могла навчилася робити певні операції та інвестувати.

Також важливо донести, що пасивний капітал не приносить нічого. З часом через інфляцію та інші можливі економічні ситуації можна втратити весь власний капітал. Наприклад, часто трапляється, що тримаючи всі кошти в одному банку чи в одній валюті, людина втрачає майже весь свій капітал, який збирала для купівлі квартири чи машини. Через це важливо кожному отримати базові навички в диверсифікації та інвестування в цілому.

В даній системі потрібно сформулювати такі економічні умови, які будуть максимально подібні до реальних речей в світі.

Зробити таку систему, щоб в майбутньому її можна було розвивати та впровадити в публічне використання. А також дати можливість розробляти інвестиційні стратегії на даних, які актуальні на даний момент часу в світі.

					ДП ІС-6127.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

В даному предметному середовищі головним суб'єктом є інвестор. Який приймає рішення та управляє своїм портфелем, тобто активами. Активи – це об'єкти предметного середовища. За допомогою різних операцій над своїм портфелем в системі, інвестор може змінювати свій прибуток, мінімізувати ризики та робити передбачення.

Інвестор, визначившись з інвестиційною стратегією, повинен придбати той чи інший актив. Він прагне отримати прибуток з цього активу. Прибуток може бути різний: відсоток від депозиту, дивіденди від акцій компаній, прибуток від різниці в ціні акцій, валюти. Все це дає інвестору можливість при правильному підході отримати прибуток та збільшити свій капітал.

Процес інвестування достатньо складний та ризикований. Не знаючи як правильно вкладати свій капітал та не дотримуючись певних правил та стратегій, можна втратити всі свої кошти.

Для того щоб повністю зрозуміти цей процес треба багато практики та знань. Проте практика в цьому на власних коштах може призвести до повної втрати своїх заощаджень.

					ДП ІС-6127.01.000 ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

В даній системі єдиним актором буде користувач, який буде взаємодіяти з системою. Користувачу потрібно буде зайти на сайт, перейти на сторінку тренажеру, далі йому потрібно буде вибрати налаштування тренажеру та запустити тренажер.

В налаштування буде входити початкова сума капіталу.

Кожен цикл користувач буде бачити свій баланс та всі його активи в портфелі з певними характеристиками. Користувач зможе переглянути графіки зміни ціни активу та його характеристики.

Відповідно користувач зможе зробити зміни в свій портфель (продати/купити) та перейти до наступного кроку циклу.

Після кожної ітерації циклу користувач отримає новий результат, який сформувався після минулих змін.

В кінці кожного кроку циклу користувач побачить рекомендації щодо змін в портфелі для максимізації прибутку у майбутньому та мінімізації ризиків втрати капіталу.

Результатом остаточного кроку буде повна ціна портфелю та зароблені кошти за весь час моделювання.

### 4.2 Вимоги до надійності

Програма повинна зберігати працездатність і забезпечувати відновлення своїх функцій при виникненні непланових ситуацій. А саме при помилці на сервері користувач повинен бути оповіщений про це та зміни внесені ним залишалися в пам'яті інтерфейсу для майбутнього використання при новій спробі звернутися до серверу. Має бути забезпечена повна відповідність між даними, які вводить користувач та тими, що потрапляють на сервер.

					ДП ІС-6127.01.000 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4.3 Вимоги до складу і параметрів технічних засобів

Склад, структура і способи організації даних в системі повинні бути визначені на етапі технічного проектування.

Структура технічних засобів визначається виходячи із можливості їх забезпечити виконання встановлених операцій процесу технічного обслуговування.

Серверна частина може бути розміщена на будь-якому сервері, який має налаштування для Інтернету, тобто, щоб інтерфейс користувача міг звернутися до нього.

Інтерфейс користувача потребує браузера для своєї роботи та виходу в інтернет, щоб робити запити на сервер.

					ДП ІС-6127.01.000 ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки системи ведення наукової роботи.

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	10.02.2020	
2.	Аналіз існуючих методів розв'язання задачі	15.03.2020	
3.	Підготовка технічного завдання на розробку програмного продукту	17.03.2020	
4.	Розробка сценарію роботи	24.03.2020	
5.	Технічне проектування – функціональність, модулі, задачі, цілі тощо	30.03.2020	
6.	Узгодження з керівником інтерфейсу користувача	15.04.2020	
7.	Розробка інформаційного забезпечення	30.04.2020	
8.	Розробка програмного забезпечення	05.05.2020	
9.	Налагодження програми	10.05.2020	
10.	Тестування програми	27.05.2020	
11.	Оформлення пояснювальної записки	30.05.2020	
12.	Подання ДП на попередній захист	15.05.2020	
13.	Подання ДП на основний захист	01.06.2020	
14.	Подання ДП рецензенту	02.06.2020	



## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ СИСТЕМИ

### 6.1 Види випробувань

Для контролю правильності роботи програмного забезпечення буде проведено функціональне тестування. В ході тестування буде проведено випробування основних функціональних характеристик системи та цілої системи загалом.

Для перевірки роботи рекомендацій та системи загалом буду проведено декілька різних прогонів з різними налаштуваннями та різними внесеними змінами під час роботи системи. Що дасть можливість перевірити на правильність роботу системи.

Також буде перевірено нетипові ситуації в системі, щоб переконатися, що система є надійною.

					ДП ІС-6127.01.000 ТЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:  
Попенко Володимир Дмитрович

ID перевірки:  
1004014823

Дата перевірки:  
13.06.2020 02:32:37 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
13.06.2020 18:44:08 EEST

ID користувача:  
77149

Назва документу: Hodarchenko\_is61\_3

ID файлу: 1004027879 Кількість сторінок: 40 Кількість слів: 5898 Кількість символів: 44413 Розмір файлу: 2.88 MB

## 11.5% Схожість

Найбільша схожість: 5.02% з джерело бібліотеки. ID файлу: 1000017749

7.6% Схожість з Інтернет джерелами

86

Page 42

9.78% Текстові збіги по Бібліотеці акаунту

253

Page 43

## 2.63% Цитат

Цитати

4

Page 44

Вилучення переліку посилань вимкнено

## 0% Вилучень

Вилучений текст відсутній

## Підміна символів

Заміна символів

11

## ВСТУП

Інвестиція, капітальні вкладення — операція, що дозволяє інвестору купити певні активи. Це, наприклад, акції компаній, валюта іноземних держав, депозити, різні види матеріальних речей.

Інвестиційна діяльність – це процес придбання та продавання активів, менеджмент ризиків та прибутку інвестиційного портфеля, який супроводжується різними стратегіями. Інвестор прагне отримати прибуток.

Інвестиційна діяльність компанії - це послідовна цілеспрямована діяльність, яка полягає в капіталізації нерухомості, купівлі та використанні інвестиційних ресурсів, регулюванні інвестиційних процесів, міжнародному русі капіталу та інвестиційних товарів, а також у створенні сприятливих умов для інвестицій і мають на меті одержання прибутку.

Важливим поняттям в інвестиційній діяльності є інвестиційні цикли. Це процес, який реалізується протягом всього часу інвестування. Цикл існує від моменту формування намірів на придбання активу до моменту отримання прибутку від його реалізації.

Об'єктами в інвестиційної діяльності можуть виступати різні активи такі, як матеріальні цінності, цінні папери та майнові права, інтелектуальні цінності. Визначені види об'єктів інвестування можуть бути як об'єктами, в які вкладаються кошти, так і формами, в яких здійснюються вкладення.

Суб'єктами інвестиційної діяльності є інвестори та учасники, які можуть бути фізичними та юридичними особами.

Інвестори — суб'єкти інвестиційної діяльності. Вони приймають рішення про інвестування в актив, а саме про вкладання власних, позичкових і залучених майнових та інтелектуальних цінностей в об'єкти інвестування. Інвестори можуть бути вкладниками, кредиторами, покупцями.

За своїм статусом інвестори можуть бути корпоративними, інституційними та індивідуальними.

ДП IC-6127.1081 ПЗ

Індивідуальні або приватні інвестори – це фізичні особи, які можуть вкладати власні нагромадження для інвестування. Індивідуальними інвесторами може бути будь-хто. Індивідуальний інвестор найчастіше зацікавлений у отриманні прибутку з грошей, які він не використовує. Він прагне досягти фінансової стабільності.

Ризики інвестиційної діяльності – це можливість отримати відмінний від очікуваного результат. Це може статися з різних причин: закриття банків, компаній, криза і так далі. Очікуваний результат – це отримання доходів або досягнення певних цілей інвестора.

Інвестор для своєї діяльності використовую різні методи та стратегії. Процес інвестування зводиться до таких кроків:

- 1) Формування інвестиційного портфелю.
- 2) Вибір стратегії управління портфелем.
- 3) Редагування портфелю.
- 4) Керування прибутком та мінімізація ризиків.

Інвестиційний портфель — набір різних активів, які інвестор може придбати. Це наприклад, цінні папери, валюта, депозити та багато інших видів. Інвестор управляє цими активами, щоб досягти певних цілей.

При формуванні портфелю інвестори використовують диверсифікацію.

Диверсифікація — це коли інвестор розподіляє свій капітал між різними фінансовими активами, щоб в разі збитків в одному з них, інші могла покрити збитки.

Дивіденди — частина прибутку, яка виплачується власникам акцій в кінці кожного місяця.

Для виконання інвестиційно-фінансової діяльності створено багато різних додатків та інформаційних систем. Проте ідеєю цієї роботи є створення тренажеру з рекомендаційною системою для навчання індивідуальних інвесторів. За допомогою цієї системи будь-яка людина зможе навчитися правильно інвестувати свої власні кошти для отримання прибутку.

					ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДП IC-6127.1081 ПЗ

Основною ідеєю є створення середовища максимально подібного до реального світового ринку (історичні дані), в якому інвестор має можливість з певною вибраною періодичністю впливати на свій портфель і дивитися можливі рекомендації по його зміні.

ДП ІС-6127.1081 ПЗ

ДП IC-6127.1081 ПЗ

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Опис предметного середовища

В даному предметному середовищі головним суб'єктом є інвестор. Який приймає рішення та управляє своїм портфелем, тобто активами. Активи – це об'єкти предметного середовища. За допомогою різних операцій над своїм портфелем в системі, інвестор може змінювати свій прибуток, мінімізувати ризики та робити передбачення.

Інвестор, визначившись з інвестиційною стратегією, повинен придбати той чи інший актив. Він прагне отримати прибуток з цього активу. Прибуток може бути різний: відсоток від депозиту, дивіденди від акцій компаній, прибуток від різниці в ціні акцій, валюти. Все це дає інвестору можливість при правильному підході отримати прибуток та збільшити свій капітал.

### 1.1.1 Опис процесу діяльності

Процес інвестування достатньо складний та ризикований. Не знаючи як правильно вкласти свій капітал та не дотримуючись певних правил та стратегій, можна втратити всі свої кошти.

Для того щоб повністю зрозуміти цей процес треба багато практики та знань. Проте практика в цьому на власних коштах може призвести до повної втрати своїх заощаджень.

### 1.1.2 Опис функціональної моделі

В даній системі єдиним актором буде користувач, який буде взаємодіяти з системою. Користувачу потрібно буде перейти на сторінку тренажеру, далі йому потрібно буде вибрати налаштування тренажеру та запустити тренажер.

В налаштування буде входити початковий капітал для вкладення. Кожен цикл користувач буде бачити свій баланс та всі його активи в портфелі з

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП ІС-6127.1081 ПЗ

ДП ІС-6127.1081 ПЗ

певними характеристиками. Користувач зможе переглянути графіки зміни ціни активу та його характеристики.

Відповідно користувач зможе зробити зміни в свій портфель (продати/купити) та перейти до наступного кроку циклу.

Після кожної ітерації циклу користувач отримає новий результат, який сформувався після минулих змін.

В кінці кожного кроку циклу користувач побачить рекомендації щодо змін в портфелі для максимізації прибутку у майбутньому та мінімізації ризиків втрати капіталу.

Результатом остаточного кроку буде повна ціна портфелю та зароблені кошти за весь час моделювання.

Таблиця 1.1 – Функціональні вимоги

Функціональні вимоги	Пріоритет
1. Система надає користувачу можливість вибрати налаштування для процесу інвестування.	Високий
2. Система дозволяє переглянути дані інвестування.	Високий
2.1. Система надає користувачу можливість переглянути власний баланс та зміни відносно останнього кроку.	Високий
2.2. Система надає користувачу можливість переглянути свій портфель, та зміни по кожному активу.	Високий
3. Система дозволяє переглянути рекомендації щодо змін в портфелі.	Високий
4. Система дозволяє купити активи.	Високий
4.1. Система дозволяє відкрити депозит.	Високий
4.2. Система дозволяє купити акції компаній.	Високий
4.3. Система дозволяє купити валюту.	Високий
5. Система дозволяє користувачу переглянути характеристики активу, що він хоче придбати.	Низький

## ДП ІС-6127.1081 ПЗ

**1.2 Огляд наявних аналогів**

Точного аналогу такої системи знайдено не було. Проте в той чи іншій мірі ця ідея реалізована в різних інформаційних системах. Наприклад, сайт investing.com при реєстрації надсилає на пошту рекомендації щодо поточної ситуації на ринку акцій. Він показує показник зміни ціни акцій за тиждень і робить припущення: ціна буде зростати і тому краще купувати чи ціна буде спадати тому краще продавати.

Існує багато різних брокерів, які дозволяють просто на демо-рахунках торгувати на біржі. Проте часто в таких системах іде розрахунок на людей, які інвестують в короткій перспективі.

Схожі системи починають впроваджуватись в банках різних країн світу. Які дозволяють переглядати власний портфель та отримувати дивіденди на власний рахунок. Гарними прикладами є «Тинькофф Инвестиции» або PayPal створили мобільний додаток для купівлі та продажу акцій. На жаль, в Україні такої можливості не існує. Для інвестування в Україні на біржі потрібні багато різних дозволів та договорів, що робить неможливим створення схожої системи на даний час.

Проте всі ці системи досить вузько направлені і не дають можливості в формі «гри» навчитися вкладати власні кошти та навчитися регулювати ризики, щоб кожен міг спробувати себе в цьому занятті.

**1.3 Постановка задачі****1.3.1 Призначення розробки**

Робота буде корисна всім хто бажає почати інвестувати та хоче на практичному прикладі навчитися розуміти графіки, реалізовувати інвестиційні стратегії, навчитися мінімізувати ризики та заробляти на власному капіталі.

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			



ДП IC-6127.1081 ПЗ

В майбутньому ця система може стати першою в Україні системою, яка буде включати в собі всі можливості інвестицій. І кожен зможе вкласти кошти у те що захоче, не витрачаючи багато часу на пошук можливості інвестувати вільні кошти по різних банкам та системам.

### 1.3.2 Цілі та задачі розробки

Створити інформаційну систему, яка буде максимально простою для використання, щоб навіть людина, яка нічого не розуміє в фінансах могла навчитися робити певні операції та інвестувати.

Також важливо донести, що пасивний капітал не приносить нічого. З часом через інфляцію та інші можливі економічні ситуації можна втратити весь власний капітал. Наприклад, часто трапляється, що тримаючи всі кошти в одному банку чи в одній валюті, людина втрачає майже весь свій капітал, який збирала для купівлі квартири чи машини. Через це важливо кожному отримати базові навички в диверсифікації та інвестування в цілому.

В даній системі потрібно сформулювати такі економічні умови, які будуть максимально подібні до реальних речей в світі.

Зробити таку систему, щоб в майбутньому її можна було розвивати та впровадити в публічне використання. А також дати можливість розробляти інвестиційні стратегії на даних, які актуальні на даний момент часу в світі.

## Висновок до розділу

Процес інвестування складний та ризиковий. Часто для отримання якогось результату інвестору доводиться втрачати кошти через свої помилки. Тому метою даного проекту є створення легкого для розуміння та використання тренажеру, щоб кожен зміг спробувати себе у цій справі.

Основною ідеєю проекту є навчання людей процесу інвестування.

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП ІС-6127.1081 ПЗ

ДП IC-6127.1081 ПЗ

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

## 2.1 Вхідні дані

Вхідні дані представлені у таблиці 2.1

Таблиця 2.1 – Вхідні дані

Назва даних	Опис
Історичні дані про ціни на акції та валюту.	Отримуються з сервісу Alpha Vantage та заносяться в базу.
Дані про налаштування процесу інвестування: початковий капітал.	Вводяться користувачем перед початком роботи системи.
Дані про сесію	Створюються під час початку роботи системи і змінюються під час її функціонування.

## 2.2 Вихідні дані

Вихідні дані представлені в таблиці 2.2.

Таблиця 2.2 – Вихідні дані

Назва даних	Опис
Рекомендації щодо зміни в портфелі	Різні рекомендації для користувача щодо покращення його портфелю.
Актуальні дані про сесію	Інформація про баланс портфелю, його склад та зміни з минулого кроку.
Дані про активи	Інформація та характеристики про активи, їх оцінка.

Змн. Арк. № докум. Підпис Дата ДП ІС-6127.1081 ПЗ Арк.

ДП ІС-6127.1081 ПЗ

2.3 Опис структури бази даних

На початку проектування бази даних інформаційної системи були виділені сутності, що відображені у відповідних таблицях бази даних (рисунок 2.1).



Рисунок 2.1 – Схема бази даних

Розглянемо таблиці та зв'язки між ними. Таблиця Currencies (рисунок 2.2) – це таблиця, які містить назви валют та їх коди. Таблиця Stocks (рисунок 2.3) – це таблиця, які містить назви акцій та їх символи.

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Code	nvarchar(MAX)	<input type="checkbox"/>
CurrencyName	nvarchar(MAX)	<input type="checkbox"/>

Рисунок 2.2 –Таблиця Currencies

ДП IC-6127.1081 ПЗ

Рисунок 2.3 – Таблица Stocks

Таблиця Sessions (рисунок 2.4) – це таблиця, яка містить інформацію про сесію. А саме її ідентифікатор та дата етапу інвестування.

### Рисунок 2.4 – Таблица Sessions

Таблиця Portfolios (рисунок 2.5) відображає портфель користувача. Вона прив'язана до таблиці Sessions і показує поточний стан вашого портфелю, а саме вільні кошти та початковий капітал.

Рисунок 2.5 – Таблица Portfolios

Таблиці CurrencyPrices (рисунок 2.6) та StockPrices (рисунок 2.7) містять історичні дані для валюти та акцій. Вони містять ціну, дату та прив'язані до таблиці Currencies та таблиці Stocks відповідно.

Рисунок 2.6 – Таблица CurrencyPrices

ДП ІС-6127.1081 ПЗ

ДП IC-6127.1081 ПЗ

Рисунок 2.7 – Таблица StockPrices

Таблиця DepositPackages (рисунок 2.8) містить інформацію про відкриті депозити, а саме такі поля: скільки коштів біло вкладено, період у місяцях, час відкриття, відсотки річних та чи виводити прибуток на рахунок кожний місяць. Ця таблиця зв'язана з таблицею Portfolios.

Рисунок 2.8 – Таблиця DepositPackages

Таблиці CurrencyPackages (рисунок 2.9) та StocksPackages (рисунок 2.10) містять поточну інформацію про поточну ціну валюти та акцій та їх кількість в портфелі. Вони зв'язані з таблицею Portfolios та таблицями Currencies та таблиці Stocks відповідно.

Рисунок 2.9 – Таблица CurrencyPackages

ДП ІС-6127.1081 ПЗ

ДП ІС-6127.1081 ПЗ

	Column Name	Data Type	Allow Nulls
PK	StockPackageId	int	<input type="checkbox"/>
	PortfolioId	int	<input type="checkbox"/>
	StockId	int	<input type="checkbox"/>
	Price	float	<input type="checkbox"/>
	Count	int	<input type="checkbox"/>

Рисунок 2.10 –Таблиця StocksPackages

Висновок до розділу

У розділі були описані вхідні та вихідні дані. Описана структура бази даних та пояснення до таблиці.

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ДП IC-6127.1081 ПЗ

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1 Змістовна постановка задачі

Для реалізації програмного забезпечення були виділені наступні задачі:

- оптимізація портфелю користувача – максимізація прибутку від портфелю та мінімізація ризиків;
- обґрунтування знайдених покращень та створення рекомендацій для користувача.

Користувач має портфель, який містить різні види активів. Актив «акції» складається з кількості акцій та їх ціни. Актив «валюта» складається з кількості валюти та її ціни до долара. «Депозит» складається з суми вкладу та місячного доходу. Першою підзадачею даної системи є підбір таких значень кількості в активах, щоб максимізувати прибуток та другою підзадачею є підбір таких значень кількості, щоб мінімізувати ризики.

### 3.2 Математична постановка задачі

Портфель складається з активів:

$$\begin{aligned} S_i &+ \dot{\zeta} \sum_{i=1}^{n_d} D_i, \\ C_i &+ \dot{\zeta} \sum_{i=1}^{n_c} \dot{\zeta} \\ P &= \sum_{i=1}^{n_c} \dot{\zeta} \end{aligned} \quad (1)$$

де  $C$  – валюта,  $S$  – акція компанії,  $D$  – відкритий депозит.  $A$  –  $n_c$ ,  $n_s$ ,  
та  $n_d$  – відповідно кількість типів валюти, акцій та депозитів.

Формула валюты:

$$\underline{C}_i = x_i \cdot l_i, i = 1, \dots, n_c, \quad (2)$$

де  $x$  – кількість купленої валюти, а  $l$  – ціна відносно долара.

Формула акції:

$$S_i = x_i \cdot l_i + m_i, i = 1', n_s, \quad (3)$$

Змн. Арк. № докум. Підпис Дата ДП ІС-6127.1081 ПЗ Арк

## ДП ІС-6127.1081 ПЗ

де  $x$  – кількість одиниць певної акції,  $l$  – ціна в доларах за одну,  $m$  – дивіденди.

Формула депозиту:

$$D_i = \frac{x_i}{12}, i=1, \dots, n_d, \quad (4)$$

де  $x$  – кількість вкладених грошей.

Задачею є знаходження таких  $x_i$ , щоб максимізувати прибуток та мінімізувати ризики.

### 3.3 Обґрунтування методу розв'язання

Введемо функцію ризиків. Вона може бути будь-якою, бо для розв'язання задачі це не так важливо. В даній системі було вирішено використати таку функцію ризиків:

$$R(A) = \frac{\sum_i^l \omega_i \cdot U_i}{\sum_i^q \omega_i \cdot I_i} \quad (5)$$

Це зважена сума збитків( $U$ ) поділена на зважену суму доходів( $I$ ).  $\omega_i$  - вага доходу чи збитку.

$$\sum \omega_i = 1 \quad (6)$$

Ці ваги будуються по принципу: більш близькі події важливіші за старі події.

Функція  $R(A)$  може бути більше нуля, тоді кількість збитків більша за кількість доходів, тому робимо висновок, що актив досить ризиковий, а може бути менше нуля, що свідчить про достатньо високу імовірність отримати прибуток.

Тоді маємо функцію, яку потрібно мінімізувати.

$$R_i \cdot x_i + \sum_{i=n_c}^{n_c+n_s} R_i \cdot x_i \rightarrow \min \quad (7)$$

$$\sum_{i=1}^{n_c} \omega_i$$

									ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата						



ДП ІС-6127.1081 ПЗ

Для неї маємо систему обмежень:

$$x_i \geq 0.8 \cdot x_i' \cdot R_i$$

$$x_i \leq 1.2 \cdot x_i' \cdot R_i$$

$$\sum_{i=1}^{n_1} l_i \cdot x_i \leq 0.2 \cdot F$$

$$\sum_{i=n_1+1}^{n_1+n_2} l_i \cdot x_i \leq 0.2 \cdot F$$

(8)

де  $x_i'$  - минула кількість активів, а  $F$  – це повний капітал. Останні два обмеження – це правило диверсифікації, яке не дає купити на всі кошти активи одного виду. Інші два обмеження – це обмеження на максимальну кількість куплених та проданих одиниць.

Друга підзадача зводиться до максимізації прибутку. Введемо поняття зважена середня ціна. Це ціна на актив помножена на ваги і поділена на суму ваг. Позначимо цю ціну, як  $G$ . Тоді маємо функцію, яку потрібно максимізувати.

$$\begin{aligned} & \sum_{i=n_1+1}^{n_1+n_2} x_i / 12 \rightarrow \max, \\ & ((\tilde{c}_i i - l_i) * x_i) + \sum_{i=n_1+1}^{n_1+n_2} \tilde{c}_i \\ & ((\tilde{c}_i i - l_i + m_i) * x_i) + \sum_{i=n_1}^{n_1+n_2} \tilde{c}_i \end{aligned} \quad (9)$$

де  $\tilde{c}_i$  – різниця між зваженою середньою ціною та новою ціною.

Перша сума цієї функції – це для акцій, друга – для валюти та  $З$  – для депозиту.

Ця функція дає нам результат у вигляді місячного прибутку, який потрібно максимізувати.

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ДП IC-6127.1081 ПЗ

Обмеження використаємо такі самі, що і в минулій функції. Крім обмежень на депозит, тут просто будемо вважати, щоб вклад був менше за 20 відсотків від повної вартості капіталу.

Отримані значення змінних треба привести до рекомендацій за такими правилами: якщо невідома змінна зменшилась треба повідомити користувача про те, що актив не прибутковий і краще продати частину цього активу, якщо ж збільшилась – порадити придбати ще одиниць цього активу.

### 3.4 Опис методів розв'язання

Для знаходження розв'язку двох цих функцій будемо використовувати симплекс метод [1].

$$C^T x \rightarrow \max (\min) \quad (10)$$

$$Ax \leq b \quad (11)$$

В нашому випадку є функції, які ми оптимізуємо, обмеження також є. Залишилося розв'язати задачі та обґрунтувати знайдені результати.

Виконаємо вимоги стандартної форми запису задачі лінійного програмування. Для цього перетворимо всі обмеження в рівності з невід’ємною правою частиною і забезпечимо невід’ємність усіх змінних.

Побудуємо і заповнимо початкову симплекс таблицю, яка є зручним інструментом для подання канонічної форми лінійної задачі. Для заповнення початкової симплекс таблиці необхідно переписати цільову функцію  $F$  у вигляді, аналогічному до системи обмежень (рисунк 3.1).

Проведемо перевірку на оптимальність – якщо всі коефіцієнти в рядку  $F$  є невід'ємними, то отриманий розв'язок є оптимальним, але якщо хоча б один коефіцієнт є від'ємний, то необхідно продовжити симплекс ітерацію, тобто заповнити наступну симплекс таблицю.

Оберемо стовпчик, в якому міститься найбільший за модулем від'ємний коефіцієнт в рядку F (ведучий стовпчик).

Оберемо ведучий рядок та ведучий елемент *arr*. Для цього скористаємось скористатись тестом мінімального відношення (MRT) –

					ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДП ІС-6127.1081 ПЗ

запишемо у відповідному рядку відношення змінної зі стовпця "план" до змінної з ведучого стовпця і визначимо мінімальне з цих відношень. Рядок з мінімальним значенням відношення буде ведучим рядком.

Модифікуємо симплекс таблицю по відношенню до ведучого елемента

$a_{rs}$ :

- для ведучого рядка:

$$\hat{a}_{rj} = \frac{a_{rj}}{a_{rs}}; \quad (12)$$

- для ведучого стовпця:

$$\hat{a}_{rs} = 1; \hat{a}_{is} = 0; j = 1, m; i \neq r; \quad (13)$$

- для решти елементів:

$$\hat{a}_{ij} = \frac{a_{ij} \cdot a_{rs} - a_{rj} \cdot a_{is}}{a_{rs}^2} \quad (14)$$

Повертаємося до перевірки на оптимальність.

Базис	План	$x_1$	$x_2$	...	$x_s$	...	$x_n$	$x_{n+1}$	...	$x_{n+m}$	MRT
F		0	$-c_1$	$-c_2$	...	$-c_s$	$-c_n$	0	...	0	$b_1$
$x_{n+1}$	$b_1$	$a_{11}$	$a_{12}$	...	$a_{1s}$	...	$a_{1n}$	1	...	0	$a_{1s}$
...											
$x_{n+r}$	$b_r$	$a_{r1}$	$a_{r2}$	...	$a_{rs}$	...	$a_{rn}$	0	...	0	$b_r$
$x_{n+m}$	$b_m$	$a_{m1}$	$a_{m2}$	...	$a_{ms}$	...	$a_{mn}$	0	...	1	$a_{ms}$

Рисунок 3.1 – Симплекс таблиця

Кінцевим результатом роботи є базисні змінні та цільова функція. Ці дані потребують обґрунтування та подальшого використання.

### Висновок до розділу

У розділі математичного забезпечення були оглянуті математичні задачі дипломного проекту. Розглянуто роботу симплекс методу та описано, як використати його в даній задачі.

											ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата								

ДП ІС-6127.1081 ПЗ

## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Програмне забезпечення складається з двох рівних частин, а саме з клієнтської та серверної.

В основному для розробки серверної частини було використано ASP.NET Core - це фреймворк з відкритим вихідним кодом, розроблений Microsoft і її спільнотою. Він є крос-платформений, а це означає що програми написані на цьому фреймворку можна запускати на будь-якій операційній системі, що дає багато можливостей на його використання.

Функціонал цього фреймворку дозволяє створити веб-застосунок, який буде незалежний від клієнтської частини, бо запити до серверу можна буде робити по http. А це дасть нам повну незалежність сервер від клієнта.

Для використання бази даних було використано **Entity Framework Core** — це найлегший спосіб доступу до бази в C#. EF Core надає можливість використовувати правило “Code First”, що дозволяє в коді створювати класи та модель системи, а потім на основі цього будується база. Розробнику не потрібно спочатку проектувати базу, а достатньо написати код для її створення. Також ця технологія дозволяє робити запити в базу, не використовуючи великі SQL-запити, що зменшує кількість коду та полегшує розробку.

Для того, щоб отримати історичні дані було використано **Alpha Vantage API**. Це REST API надає багато різних даних та характеристик, у тому числі історичних, по валютам, акціям та іншим активам.

Для розробки клієнтської частини було використано Angular, HTML, CSS, Bootstrap.

**Angular** — TypeScript front-end фреймворк з відкритим кодом. Він розробляється у компанії Google. Через те, що він дозволяє писати на TypeScript, розробка полегшується.

ДП ІС-6127.1081 ПЗ

Арк.

Змн. Арк. № докум. Підпис Дата

ДП IC-6127.1081 ПЗ

**HTML** — мова розмітки документів. Використовується як розмітка для сторінок веб-застосунку.

**CSS** — це спеціальна мова стилів, яка дозволяє додавати певні стилі для сторінок.

**Bootstrap** — це набір інструментів, які мають відкритий код, вони використовуються для розробки веб-сайтів та веб-додатків. Ці інструменти містять шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Шаблон сайту та деякі компоненти клієнта були взяті з шаблону **Argon Dashboard Angular**, яка має відкритий вихідний код та поставляється під ліцензією MIT. Ліцензія міститься в коді роботи. [3]

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Для роботи серверу необхідні наступні параметри системи:

- процесор с частотою 2.4 ГГц або більш потужний;
- ОЗУ об'ємом 4 ГБ;
- мінімум 10 Гб доступного простору на жорстким диску;
- жорсткий диск с частотою обертання 5 400 об/хв або більше.

Також необхідне встановлене наступне програмне забезпечення:

- Ubuntu 18.04 and Windows;
- MSSQL;
- ASP.NET Core Runtime 2.2.0.

Вимоги до клієнту — це наявність одного з веб-браузерів, який підтримує Angular (наприклад, Google Chrome 83+).

ДП ІС-6127.1081 ПЗ

ДП IC-6127.1081 ПЗ

### 4.3 Архітектура програмного забезпечення

### 4.3.1 Діаграма класів

На діаграмі класів (рисунку 4.3.3) наведено модель системи, яка відповідає за оновлення даних та формування даних для використання у рекомендаціях. Тут є 3 класи `CurrencyDataUpdater`, `DepositDataUpdater` та `StockDataUpdater`, які оновлюють всю інформацію. Класи `Currency`, `Stock` та `Deposit` мають методи та характеристики, які треба оновлювати кожний крок.

Рисунок 4.3.3 – Діаграма класів

### 4.3.2 Діаграма послідовності

На діаграмі послідовності показано основну роботу програми. Спочатку користувач за допомогою клієнта робить запит для того, щоб розпочати сесію. Клієнтська частина передає HTTP POST запит до серверу. Сервер створює запис в базі та повертає ідентифікатор сесії. Далі користувач попадає на сторінку сесії. Тут він може купити активи та за допомогою POST

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	ДП IC-6127.1081 ПЗ	

ДП IC-6127.1081 ПЗ

запитів додати необхідні активи в портфель, при чому сервер додає ці записи в базу.

Останнім етапом є перехід до наступного кроку. Клієнт робить POST запит, сервер дістає всі дані про портфель сесії з бази, потім дістає нові дані та формує пакет даних, який потім використовується для знаходження рекомендацій. Останнім кроком сервер оновлює базу новими даними та повертає отримані результати клієнту.

Діаграма послідовності знаходиться у графічних матеріалах.

### 4.3.3 Діаграма компонентів

На діаграмі компонент (рисунок 4.3.3) зображені ключові компоненти сервера:

- Flask Application – серверна частина проекту;
- MSSQL Database – база даних.

А також компоненти клієнта:

- Web browser – браузер користувача.

Рисунок 4.3.3 – Діаграма компонент

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ДП IC-6127.1081 ПЗ

#### 4.3.4 Специфікація функцій

Назва	Примітка
<b>Клас DataProviderService</b> – сервіс що відповідає за інформацію про валюту та акції Task<List<Currency>> GetCurrencies()	Дістає з бази назву валюти та її код
<b>Клас StocksService</b> – сервіс що відповідає за операції над акціями в портфелі Task Buy(int portfolioId, string symbol, int count) Task Sell(int portfolioId, string symbol, int count)	Дістає з бази назву акції та її символ Додає акції в портфель Видаляє акції з портфеля
<b>Клас CurrenciesService</b> – сервіс що відповідає за операції над валютою в портфелі Task Buy(int portfolioId, string code, double cash) Task Sell(int portfolioId, string code, double cash)	Додає валюту в портфель Видаляє валюту з портфеля
<b>Клас DepositService</b> – сервіс що відповідає за операції над депозитами в портфелі Task Open(int portfolioId, double cash, double interest, int period, Boolean deduceInterest)	Відкриває депозит
<b>Клас SessionService</b> – сервіс що відповідає за сесію Task<string> Create(double startCapital) Task<SessionSnapshot > NextStep(int sessionId)	Створює сесію Оновлює всі дані, створює рекомендації

## Висновок до розділу

У розділі біли розглянуті засоби розробки системи, вимоги до технічного забезпечення. А також детально розглянуто модель системи у вигляді діаграми класів та за допомогою специфікацій функцій. Та показано послідовність дій та інфраструктуру системи.

ДП IC-6127.1081 ПЗ



ДП ІС-6127.1081 ПЗ

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

На головній сторінці знаходиться форма для налаштування сесії (рисунок 5.1). Як тільки нажимається кнопка «РОЗПОЧАТИ» головна сторінка змінюється на сторінку сесії, на якій зображується основна інформація по сесії, а також рекомендації та портфель користувача (рисунок 5.2).

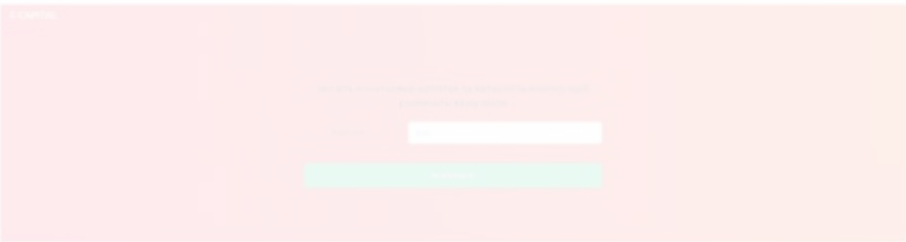


Рисунок 5.1 – Головна сторінка, поки сесія не розпочата



Рисунок 5.2 – Сторінка сесії

На сторінці сесії можливо додати активи натиснувши на кнопку «Додати новий актив». Тоді відкривається сторінка з можливими активами (рисунок 5.3).

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ДП ІС-6127.1081 ПЗ

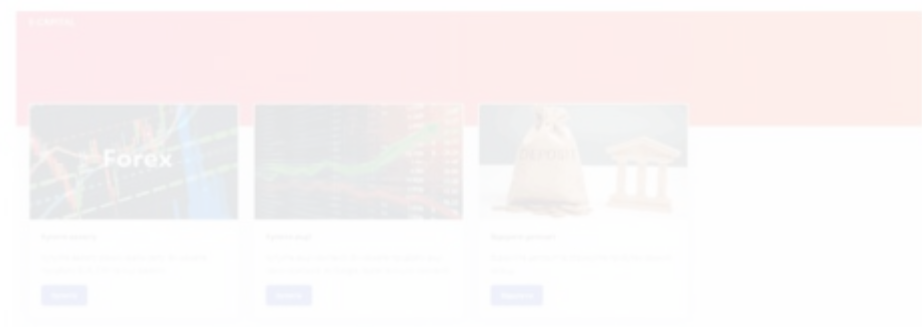


Рисунок 5.3 – Сторінка вибору активу

На сторінці вибору активу (рисунок 5.3) можна вибрати один з трьох можливих активів: депозит, акції або валюта. Після вибору необхідного активу треба натиснути відповідну кнопку.

На сторінці відкриття депозиту (рисунок 5.4) знаходяться всі необхідні налаштування, щоб відкрити депозит.



Рисунок 5.4 – Сторінка відкриття депозиту

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ДП ІС-6127.1081 ПЗ

На сторінці купівлі акцій (рисунок 5.5) можна купити або продати акції компаній, для цього необхідно зробити відповідні дії на правій та лівій картці – «Додати в кошик», «Видалити».



Рисунок 5.5 – Сторінка купівлі/продажі акцій

На сторінці купівлі валюти (рисунок 5.6) можна купити або продати валюту країн світу, для цього необхідно зробити відповідні дії на правій та лівій картці – «Додати в кошик», «Видалити» та записати кількість валюти, яку треба купити/продати.



Рисунок 5.6 – Сторінка купівлі/продажі валюти

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

## ДП ІС-6127.1081 ПЗ

## 5.2 Випробування програмного продукту

## 5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій інформаційної система з підтримки інвестиційно-фінансової діяльності вимогам технічного завдання.

## 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

## 5.2.3 Результати випробувань

Під час тестування була перевірена функціональна складова системи. Нижче наведені результати випробувань у вигляді таблиць.

Таблиця 5.1 – Початок сесії

Мета тесту	Перевірка початку сесії
Початковий стан моделі	Початкова сторінка веб-застосунку
Схема проведення тесту	Ввести початковий капітал і натиснути «РОЗПОЧАТИ».
Очікуваний результат	Екран сесії
Результат роботи	Екран сесії

Таблиця 5.2 – Додати актив

Мета тесту	Перевірка форми з додаванням активу
Початковий стан моделі	Сторінка сесії
Схема проведення тесту	Натиснути кнопку «Додати новий актив» на картці «Ваші активи»
Очікуваний результат	Екран вибору активу
Результат роботи	Екран вибору активу

Арк.

ДП ІС-6127.1081 ПЗ

Змн. Арк. № докум. Підпис Дата

### Таблиця 5.3 – Відкрити депозит

Мета тесту	Перевірка можливості відкриття депозиту
Початковий стан моделі	Сторінка відкриття депозиту
Схема проведення тесту	Ввести кількість грошей, які потрібно покласти, вибрати кількість місяців, вибрати чи додавати відсотки до рахунку і натиснути кнопку «Відкрити»
Очікуваний результат	Закриття екрану з депозитом і повернення на екран сесії
Результат роботи	Закриття екрану з депозитом і повернення на екран сесії, депозит з'явився в списку наявних активів

**Таблиця 5.4 – Купити акції/валюту**

Мета тесту	Перевірка можливості купівлі акцій/валюти
Початковий стан моделі	Сторінка акцій/валюти
Схема проведення тесту	На картці «Акції компаній»/«Валюти» біля необхідної акції/валюти натиснути кнопку «Додати в кошик». Натиснути кнопку «Придбати»/ввести кількість валюти на натиснути кнопку «Придбати».
Очікуваний результат	Закриття екрану з акціями/валютою і повернення на екран сесії
Результат роботи	Закриття екрану з акціями/валютою і повернення на екран сесії, відповідні активи з'явилися в списку наявних активів

**Таблиця 5.5 – Перехід до наступного кроку**

Мета тесту	Перевірка переходу до наступного кроку
Початковий стан моделі	Сторінка сесії
Схема проведення тесту	Натиснути кнопку «Наступний крок»

ДП IC-6127.1081 ПЗ

Арк.

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП ІС-6127.1081 ПЗ

Очікуваний результат	Сторінка завантаження, а потім сторінка сесії
Результат роботи	Сторінка завантаження, а потім сторінка сесії. На новій сторінці змінились дані, додалися рекомендації.

Таблиця 5.6 – Перегляд даних про акцію/валюту

Мета тесту	Перевірка сторінки з інформацією про акцію/валюту
Початковий стан моделі	Сторінка акцій/валюти
Схема проведення тесту	На картці «Акції компаній»/«Валюти» натиснути на необхідну акцію/валюту
Очікуваний результат	Сторінка інформації з графіками та даними
Результат роботи	Сторінка інформації з графіками та даними

**Висновок до розділу**

У розділі було розглянуто інструкцію користувача, ознайомлення з можливостями системи. Було успішно виконане тестування головних функцій системи.

**ЗАГАЛЬНІ ВИСНОВКИ**

У сучасному світі інвестування має досить велике значення, адже кожен прагне зберегти свої гроші та прибільшити їх. Але процес інвестування складний та ризиковий. За неправильного інвестування можна втратити всі кошти. Тому ідеєю цієї роботи є створення тренажеру, який дозволить спробувати себе у інвестуванні і навчитися цьому процесу.

В роботі розглянуто математичну постановку задачі, а саме математику для рекомендацій. Побудовано вимоги до системи та розроблено тести для неї.

Система була створена достатньо гнучкою для того, щоб у майбутньому

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

Ця робота буде корисна для ознайомлення будь-кому, хто захоче спробувати себе у інвестуванні на реальних прикладах із життя.

					ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

ДП ІС-6127.1081 ПЗ

ПЕРЕЛІК ПОСИЛАНЬ

- 1. Хемди А. Таха Глава 3. Симплекс-метод. Введение в исследование операций. 7-е изд. М.: "Вильямс", 2007.
- 2. Томас Х. Кормен и др. Глава 29. Линейное программирование. Алгоритмы: построение и анализ. 2-е изд. М.: «Вильямс», 2006. С. 1296. ISBN 5-8459-0857-4.
- 3. Argon Dashboard Angular [Электронный ресурс] – Режим доступа до ресурсу:  
<https://github.com/creativetimofficial/argon-dashboard-angular.git>.

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			



ДП ІС-6127.1081 ПЗ

Додаток А

**Тексти програмного коду**  
**Інформаційна система з підтримки інвестиційно-фінансової діяльності**  
(Найменування програми (документа))

DVD-R  
(Вид носія даних)  
8 арк, 244 Кб  
(Обсяг програми (документа) , арк.,) Кб)

КИЇВ – 2020 РОКУ

						ДП ІС-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

ДП IC-6127.1081 ПЗ

### Лістинг основної частини програми

```

public abstract class Asset
{
    public abstract AssetType Type { get; }
    public decimal Risk { get; protected set; }
    public decimal WeightedAverage { get; protected set; }
    public decimal LastUpdateIncome { get; protected set; }
}

public enum AssetType
{
    Deposit = 0,
    Currency = 1,
    Stock = 2
}

public interface IDataUpdater
{
    Task Update(Asset asset);
}

public class Currency : Asset
{
    public override AssetType Type => AssetType.Currency;

    public string FromSymbol { get; }
    public string ToSymbol => "USD";

    public decimal Price { get; private set; }
    public decimal Value { get; private set; }
    public decimal Sum { get; private set; }

    public DateTime LastUpdateTime { get; private set; }

    public void UpdateTime()
    {
        LastUpdateTime = LastUpdateTime.AddMonths(1);
    }

    public void Add(decimal value)
    {
        Value += value;
    }

    public void ChangePrice(decimal newPrice)
    {
        Price = newPrice;
    }

    public void ChangeSum()
    {
        Sum = Value * Price;
    }
}

public class CurrencyDataUpdater : IDataUpdater

```

Арк.

ДП IC-6127.1081 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП ІС-6127.1081 ПЗ

```
private IMonthlyCurrencyDataService _monthlyCurrencyDataService;

public CurrencyDataUpdater()
{
    _monthlyCurrencyDataService = new MonthlyCurrencyDataService(new ApiClient());
}

public async Task Update(Asset asset)
{
    var currency = asset as Model.Currency;

    var data = await _monthlyCurrencyDataService.GetData(currency.FromSymbol, currency.ToSymbol,
        CancellationToken.None);

    currency.UpdateTime();

    var newData = data.MonthlyTimeSeries.FirstOrDefault(o => o.Key == currency.LastUpdatedTime);
    currency.ChangePrice(decimal.Parse(newData.Value.Close));
    currency.ChangeSum();
}
}

public class Deposit : Asset
{
    public decimal Value { get; }
    public decimal Income { get; private set; }
    public decimal Interest { get; }
    public bool DeduceInterest { get; }
    public int MonthToHold { get; }
    public int MonthCompleted { get; private set; }

    public bool MustBeClosed => MonthToHold == MonthCompleted;

    public override AssetType Type => AssetType.Deposit;

    public Deposit(decimal value, decimal interest, bool deduceInterest, int monthToHold)
    {
        Interest = interest;
        Value = value;
        DeduceInterest = deduceInterest;
        MonthToHold = monthToHold;
    }

    internal void AddIncome(decimal value)
    {
        MonthCompleted++;
        Income += value;
        if (DeduceInterest)
        {
            LastUpdateIncome = value;
        }
    }
}

public class DepositDataUpdater : IDataUpdater
```

Арк.

ДП IC-6127.1081 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП IC-6127.1081 ПЗ

```
private readonly IDepositOperations _depositOperations;

public DepositDataUpdater()
{
    _depositOperations = null;
}

public Task Update(Asset asset)
{
    var deposit = asset as Model.Deposit;
    var income = _depositOperations.CalculateMonthIncome(deposit.Value, deposit.MonthToHold,
deposit.Interest);
    deposit.AddIncome(income);
    return Task.CompletedTask;
}

public class Stocks : Asset
{
    public override AssetType Type => AssetType.Stock;

    public int Count { get; private set; }
    public string Symbol { get; private set; }
    public decimal Price { get; private set; }
    public decimal MonthDividends { get; }
    public decimal Sum { get; private set; }

    public DateTime LastUpdateTime { get; private set; }

    public Stocks(string symbol, int count, decimal price)
    {
        LastUpdateTime = new DateTime(2010, 1, 1);
        Symbol = symbol;
        Price = price;
        Count = count;
        MonthDividends = LastUpdateIncome = new Random().Next(5, 20);
    }

    public void UpdateTime()
    {
        LastUpdateTime = LastUpdateTime.AddMonths(1);
    }

    public void AddStocks(int count)
    {
        Count += count;
    }

    public void ChangePrice(decimal newPrice)
    {
        Price = newPrice;
    }

    public void ChangeSum()
    {
        Sum = Count * Price;
    }
}
```

							ДП IC-6127.1081 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата				

ДП IC-6127.1081 ПЗ

```
public class StockDataUpdater : IDataUpdater
{
    private MonthlyStockDataService _monthlyStockDataService;

    public StockDataUpdater()
    {
        _monthlyStockDataService = new MonthlyStockDataService(new ApiClient());
    }

    public async Task Update(Asset asset)
    {
        var stock = asset as Model.Stocks;
        var data = await _monthlyStockDataService.GetData(stock.Symbol, CancellationToken.None);

        stock.UpdateTime();

        var newData = data.MonthlyTimeSeries.FirstOrDefault(o => o.Key == stock.LastUpdatedTime);
        stock.ChangePrice(decimal.Parse(newData.Value.Close));
        stock.ChangeSum();
    }
}

public class MonthlyCurrencyData
{
    [JsonProperty("Meta Data")]
    public MetaData MetaData { get; set; }

    [JsonProperty("Time Series FX (Monthly)")]
    public Dictionary<DateTime, MonthlyTimeSery> MonthlyTimeSeries { get; set; }
}

public class MetaData
{
    [JsonProperty("1. Information")]
    public string Information { get; set; }

    [JsonProperty("2. From Symbol")]
    public string FromSymbol { get; set; }

    [JsonProperty("3. To Symbol")]
    public string ToSymbol { get; set; }

    [JsonProperty("4. Last Refreshed")]
    public DateTimeOffset LastRefreshed { get; set; }

    [JsonProperty("5. Time Zone")]
    public string TimeZone { get; set; }
}

public class MonthlyTimeSery
{
    [JsonProperty("1. open")]
    public string Open { get; set; }

    [JsonProperty("2. high")]
    public string High { get; set; }

    [JsonProperty("3. low")]
    public string Low { get; set; }
}
```

Арк.

ДП IC-6127.1081 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП IC-6127.1081 ПЗ

```
[JsonProperty("4. close")]

    public string Close { get; set; }
}

public interface IMonthlyCurrencyDataService
{
    Task<MonthlyCurrencyData> GetData(string fromSymbol, string toSymbol, CancellationToken
cancellationToken);
}

public class MonthlyCurrencyDataService : IMonthlyCurrencyDataService
{
    private readonly IApiClient _apiClient;
    private const string PARAM_SEPARATOR = "&";
    private const string FUNCTION = "function=FX_MONTHLY";

    public MonthlyCurrencyDataService(IApiClient apiClient)
    {
        _apiClient = apiClient;
    }

    public Task<MonthlyCurrencyData> GetData(string fromSymbol, string toSymbol, CancellationToken
cancellationToken)
    {
        return _apiClient.Get<MonthlyCurrencyData>(CreateParams(fromSymbol, toSymbol), cancellationToken);
    }

    private string CreateParams(string fromSymbol, string toSymbol)
    {
        return $"{FUNCTION}{PARAM_SEPARATOR}from_symbol={fromSymbol}
{PARAM_SEPARATOR}to_symbol={toSymbol}";
    }
}

public interface IApiClient
{
    Task<TResponse> Get<TResponse>(string parametr, CancellationToken cancellationToken);
}

public class ApiClient : IApiClient
{
    private const string PARAM_SEPARATOR = "?";
    private const string ACCESS_TOKEN = "&apikey=";

    public async Task<TResponse> Get<TResponse>(string parametr, CancellationToken cancellationToken)
    {
        using (var client = new HttpClient() { BaseAddress = new Uri("https://www.alphavantage.co/query") })
        {
            var response = await client.GetAsync(UrlFormatter(parametr));
            response.EnsureSuccessStatusCode();
            var content = await response.Content.ReadAsStringAsync();
            return JsonConvert.DeserializeObject<TResponse>(content);
        }
    }

    private string UrlFormatter(string parametr)
```

Арк.

ДП IC-6127.1081 ПЗ

Змн. Арк. № докум. Підпис Дата

ДП IC-6127.1081 ПЗ

```

return $"{{PARAM_SEPARATOR}}{params}{{ACCESS_TOKEN}}";
    }
}

public class AnalyzerService : IAnalyzerService
{
    public Task<List<string>> GetRecommendation(decimal total, List<Deposit> deposits, List<Stocks> stocks,
List<Currency> currencies)
    {
        List<string> recommendations = new List<string>();
        recommendations.AddRange(GetRiskMinimizationRecommendations(total, stocks, currencies));
        recommendations.AddRange(GetIncomeMaximizationRecommendations(total, stocks, currencies,
deposits));

        return Task.FromResult(recommendations);
    }

    private List<string> GetRiskMinimizationRecommendations(decimal total, List<Stocks> stocks,
List<Currency> currencies)
    {
        var variablesCount = stocks.Count + currencies.Count;

        var constraints = new List<Constraint>();
        constraints.AddRange(GetStockConstraints(total, variablesCount, stocks));
        constraints.AddRange(GetCurrencyConstraints(total, variablesCount, stocks.Count * 2 + 1, currencies));

        var function = GetMinimizeFunction(stocks, currencies);

        var simplexAlgorithm = new SimplexAlgorithm(function, constraints.ToArray());
        var result = simplexAlgorithm.GetResult();

        return new List<string>();
    }

    private List<string> GetIncomeMaximizationRecommendations(decimal total, List<Stocks> stocks,
List<Currency> currencies, List<Deposit> deposits)
    {
        var variablesCount = stocks.Count + currencies.Count;

        var constraints = new List<Constraint>();
        constraints.AddRange(GetStockConstraints(total, variablesCount, stocks));
        constraints.AddRange(GetCurrencyConstraints(total, variablesCount, stocks.Count * 2 + 1, currencies));
        constraints.AddRange(GetDepositConstraints(total, variablesCount,
            stocks.Count * 2 + currencies.Count * 2 + 2, deposits));

        var function = GetMaximizeFunction(stocks, currencies, deposits);

        var simplexAlgorithm = new SimplexAlgorithm(function, constraints.ToArray());
        var result = simplexAlgorithm.GetResult();

        return new List<string>();
    }
}

```

Арк.

ДП IC-6127.1081 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ДП IC-6127.1081 ПЗ

```
private Function GetMaximizeFunction(List<Stocks> stocks, List<Currency> currencies, List<Deposit> deposits)
{
    double[] variables = new double[stocks.Count + currencies.Count + deposits.Count];
    var index = 0;
    foreach (var stock in stocks)
    {
        variables[index] = (double)(stock.WeightedAverage - stock.Price + stock.MonthDividends);
        index++;
    }
    foreach (var currency in currencies)
    {
        variables[index] = (double)(currency.WeightedAverage - currency.Price);
        index++;
    }
    foreach (var deposit in deposits)
    {
        variables[index] = (double)(deposit.Interest / 12);
        index++;
    }

    return new Function(variables, 0, false);
}

private Function GetMinimizeFunction(List<Stocks> stocks, List<Currency> currencies)
{
    double[] variables = new double[stocks.Count + currencies.Count];
    var index = 0;
    foreach (var stock in stocks)
    {
        variables[index] = (double)stock.Risk;
        index++;
    }
    foreach (var currency in currencies)
    {
        variables[index] = (double)currency.Risk;
        index++;
    }

    return new Function(variables, 0, false);
}

private Constraint[] GetStockConstraints(decimal total, int variablesCount, List<Stocks> stocks)
{
    var constraints = new List<Constraint>();
    var index = 0;
    foreach (var stock in stocks)
    {
        var variables = new double[variablesCount];
        variables[index] = 1;
        constraints.Add(new Constraint(variables, 0.8 * stock.Count, ">="));
        constraints.Add(new Constraint(variables, 1.2 * stock.Count, "<="));
        index++;
    }

    var divVariables = new double[variablesCount];
```

Арк.

ДП IC-6127.1081 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------



ДП IC-6127.1081 ПЗ

```

index = 0;

        foreach (var stock in stocks)
        {
            divVariables[index] = (double)stock.Price;
            index++;
        }
        constraints.Add(new Constraint(divVariables, 0.4 * (double)total, "<="));

        return constraints.ToArray();
    }

    private Constraint[] GetCurrencyConstraints(decimal total, int variablesCount, int startIndex, List<Currency>
currencies)
    {
        var constraints = new List<Constraint>();
        var index = startIndex;
        foreach (var currency in currencies)
        {
            var variables = new double[variablesCount];
            variables[index] = 1;
            constraints.Add(new Constraint(variables, 0.8 * (double)currency.Value, ">="));
            constraints.Add(new Constraint(variables, 1.2 * (double)currency.Value, "<="));
            index++;
        }

        var divVariables = new double[variablesCount];
        index = startIndex;
        foreach (var currency in currencies)
        {
            divVariables[index] = (double)currency.Price;
            index++;
        }
        constraints.Add(new Constraint(divVariables, 0.3 * (double)total, "<="));

        return constraints.ToArray();
    }

    private Constraint[] GetDepositConstraints(decimal total, int variablesCount, int startIndex, List<Deposit>
deposits)
    {
        var constraints = new List<Constraint>();
        var index = startIndex;
        var divVariables = new double[variablesCount];
        foreach (var deposit in deposits)
        {
            divVariables[index] = 1;
            index++;
        }
        constraints.Add(new Constraint(divVariables, 0.3 * (double)total, "<="));

        return constraints.ToArray();
    }
}

public interface IAnalyzerService
{
    Task<List<string>> GetRecommendation(decimal total,
        List<Deposit> deposits, List<Stocks> stocks, List<Currency> currencies);
}

```

Арк.

ДП IC-6127.1081 ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------





1	Melnichuk_bachelor	ID файлу: 1000017749	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	44 Джерело	5.02%
2	Lubivij_bachelor	ID файлу: 1000044105	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	4.61%
3	Bober_bachelor	ID файлу: 1000018789	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	4 Джерело	4.61%
4	Dorogin_bachelor	ID файлу: 1000037718	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	4.58%
5	Kosjak_bachelor	ID файлу: 1000017966	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	4.53%
6	+Pivovarenko_is61	ID файлу: 1003963098	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	4.51%
7	Kovalenko_bachelor	ID файлу: 5859077	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	9 Джерело	4.44%
8	Osinnij_bachelor	ID файлу: 5882681	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	3 Джерело	4.44%
9	Gychkevych_bachelor	ID файлу: 1000017950	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	2 Джерело	4.41%
10	Rastvorova_bachelor	ID файлу: 1000017748	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	7 Джерело	4.37%
11	Youssef_bachelor_IS-z51	ID файлу: 12273022	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.32%
12	Mihajlenko	ID файлу: 5850223	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.32%
13	Logvinovskij_bachelor_isz51	ID файлу: 1000070109	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.27%
14	Migal_bachelor	ID файлу: 1000020491	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.26%
15	Klimenko_Voloshin_bachelor	ID файлу: 5824717	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	2 Джерело	4.26%
16	Golubec_bachelor	ID файлу: 1000017968	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.26%
17	Kaminskij_bachelor	ID файлу: 1000044112	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.26%
18	Stepovyi_bachelor_ISz51	ID файлу: 1000017785	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.2%
19	-Sivachenko_isz61	ID файлу: 1003994454	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.15%
20	Nabokov_bachelor	ID файлу: 5882670	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.15%
21	Akimov_bachelor	ID файлу: 12272832	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		4.05%
23	Lukova_bachelor	ID файлу: 1003798235	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		3.98%

24	Klimenko_bachelor	ID файлу: 5824711	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	2 Джерело	3.9%
26	Romanchenko_bachelor	ID файлу: 5857472	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	6 Джерело	3.66%
28	Tkachenko_bachelor_isz51	ID файлу: 1000094004	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	33 Джерело	3.24%
30	Kazmirchuk_bachelor	ID файлу: 12191135	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.37%
31	Cherkov_bachelor	ID файлу: 1000044113	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.24%
32	Medvednikov_bachelor	ID файлу: 1000017739	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.2%
33	Madojan_Shishman_bachelor	ID файлу: 1000017976	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.19%
34	Potilchak_bachelor	ID файлу: 12272972	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.14%
35	Prokopenko_bachelor	ID файлу: 1000017741	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.07%
37	Pidgulko_bachelor_isz41	ID файлу: 1000066113	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		1.03%
42	Студентська робота	ID файлу: 12271359	Institution: National University of Water Management and Natural Resources	41 Джерело	0.73%
44	Kosticheva_Zinchenko_bachelor	ID файлу: 5849537	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	33 Джерело	0.51%
45	Severcev_bachelor_is62	ID файлу: 1003935272	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.49%
47	+Kozlova_is61	ID файлу: 1003963121	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.39%
48	Студентська робота	ID файлу: 6025923	Institution: Lviv Polytechnic National University		0.39%
50	Lavrinovich_bachelor	ID файлу: 5962508	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.37%
51	Polupan_bachelor	ID файлу: 5839485	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	13 Джерело	0.32%
52	Davidenko_bachelor_v2	ID файлу: 6030842	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.31%
53	Butusov_bachelor	ID файлу: 5857457	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.29%
55	Fomin_bachelor	ID файлу: 1003798192	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.27%
56	Musienko_bachelor_is62_2	ID файлу: 1003947692	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.27%
57	[UPD]Дисертація Акименко А. КМ-81МП - скорочена	ID файлу: 1000795870	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"		0.25%

58	Dmitrjuk_bachelor_IP43_v3	ID файлу: 6003163	Institution: National Technical University of Ukraine "Kyiv Polytechni...	0.2%
59	Студентська робота	ID файлу: 3042797	Institution: Lviv Polytechnic National University	6 Джерело 0.2%
60	Oshijko_magistr_IP	ID файлу: 1003315753	Institution: National Technical University of Ukraine "Kyiv Polytec	3 Джерело 0.17%
62	Студентська робота	ID файлу: 12250913	Institution: National University of Life and Environmental Sciences of U...	0.17%
63	+Lopata_is63	ID файлу: 1003963111	Institution: National Technical University of Ukraine "Kyiv Polytechnic Institute"	0.17%
64	Мельник2	ID файлу: 1000044473	Institution: National Technical University of Ukraine "Kyiv Polytechnic Ins	2 Джерело 0.15%
70	+Proskurka_is61_3	ID файлу: 1003994452	Institution: National Technical University of Ukraine "Kyiv Polytechnic Inst...	0.14%

## Цитати

Цитати

4

- 1 3.4 **Опис методів розв'язання** Для знаходження розв'язку двох цих функцій будемо використовувати симплекс метод [1].  $cTx \rightarrow \max (\min) (10) Ax \leq b (11)$  В нашому випадку є функції, які ми оптимізуємо, обмеження також є.
- 2 `Deposit; public Deposit(decimal value, decimal interest, bool deduceInterest, int monthToHold) { Interest = interest; Value = value; DeduceInterest = deduceInterest; MonthToHold = monthToHold; } internal void AddIncome(decimal value) { MonthCompleted++; Income += value; if (DeduceInterest) { LastUpdateIncome = value; } } }` public class DepositDataUpdater : IDataUpdater 33 **ДП ІС-6127.1081 ПЗ Змн.**
- 3 `Stock; public int Count { get; private set; } public string Symbol { get; private set; } public decimal Price { get; private set; } public decimal MonthDividends { get; } public decimal Sum { get; private set; } public DateTime LastUpdatedTime { get; private set; } public Stocks(string symbol, int count, decimal price) { LastUpdatedTime = new DateTime(2010, 1, 1); Symbol = symbol; Price = price; Count = count; MonthDividends = LastUpdateIncome = new Random().`
- 4 `Interest / 12); index++; } return new Function(variables, 0, false); } private Function GetMinimizeFunction(List<Stocks> stocks, List<Currency> currencies) { double[] variables = new double[stocks.`

# **Графічний матеріал до дипломного проєкту**

на тему: Інформаційна система з підтримки інвестиційно-фінансової  
діяльності

---

Київ – 2020 року



**E-CAPITAL**

Введіть початковий капітал та натисніть кнопку, щоб розрахувати вашу ставку

Капітал:

**РАЗОРАТИ**

**E-CAPITAL**

**Поточний курс**

Курсова ставка: **9300 \$** ↑ 0.5%  
Потенціал вашої торгівлі

Дивиденд: **+ 54 \$** ↑ 0.4%  
3 місячного курсу

Потенційний дохід: **2500 \$** ↑ 1.2%  
3 місячного курсу інвестування

Вартість кошти: **5000 \$** ↑ 0.1%  
Тільки на вашому рахунок

**Ваші активи**

Код активу	Курсова ставка	Кількість акцій	Вартість акцій
Bitcoin (BTC)	40000	125	5000
Facebook (FB)	20000	125	2500
Apple (AAPL)	30000	100	3000
Bank of America (BAC)	3000	100	300

**Рекомендації**

Продати 50 акцій компанії Facebook, щоб зменшити ризик на 10%

Купити 5 акцій Apple, щоб збільшити потенціал на 10%

**E-CAPITAL**

**Forex**

Купити валюту  
Купити валюту з різних країн світу. Ви можете торгувати EUR, USD та інші валюти.

**Купити акції**  
Купити акції компаній. Ви можете торгувати акції різних компаній на Google, Apple та інших компаніях.

**Вкладити депозит**  
Вкладити депозит та отримати прибуток відносно входу.

**E-CAPITAL**

**Акції**

Акції — це частини компанії, що дають їй право голосу та частину прибутку. Вони є одним з основних способів інвестування грошей. Купити акції компанії, щоб збільшити свій прибуток. Вони є одним з основних способів інвестування грошей. Купити акції компанії, щоб збільшити свій прибуток.

Актив	Курсова ставка	Кількість акцій	Вартість акцій
Facebook	20000	27	5400
MSD	1000	20	2000

**Акції компанії**

Купити акції компанії, щоб збільшити свій прибуток. Вони є одним з основних способів інвестування грошей. Купити акції компанії, щоб збільшити свій прибуток.

Ціна акцій компанії: 200  
Ваша ставка: 5000  
**Продати**

**E-CAPITAL**

**Депозит**

Депозит — це вкладення грошей на певний термін. Це один з основних способів інвестування грошей. Купити депозит, щоб збільшити свій прибуток. Вони є одним з основних способів інвестування грошей.

**Накладування**

Скільки грошей вкладати:

На скільки місяців вкладати:

Процент, %:

Чи робити з прибутком?  
☐ Виплатити на рахунок  
☒ Вкласти до строку депозиту

**DEPOSIT**

Ви отримаєте прибуток після закінчення строку вкладення: **13 (USD \$)**

Зверніть увагу, що депозит не можна буде закрити раніше строку. Тому якщо ви не можете вкласти гроші, ви можете отримати прибуток 5 місяців.

**Вкласти**

**E-CAPITAL**

**Валюта**

Валюта — це грошова одиниця, що використовується в країні. Це один з основних способів інвестування грошей. Купити валюту, щоб збільшити свій прибуток. Вони є одним з основних способів інвестування грошей.

Купити валюту з різних країн світу. Ви можете торгувати EUR, USD та інші валюти.

**Валюта**

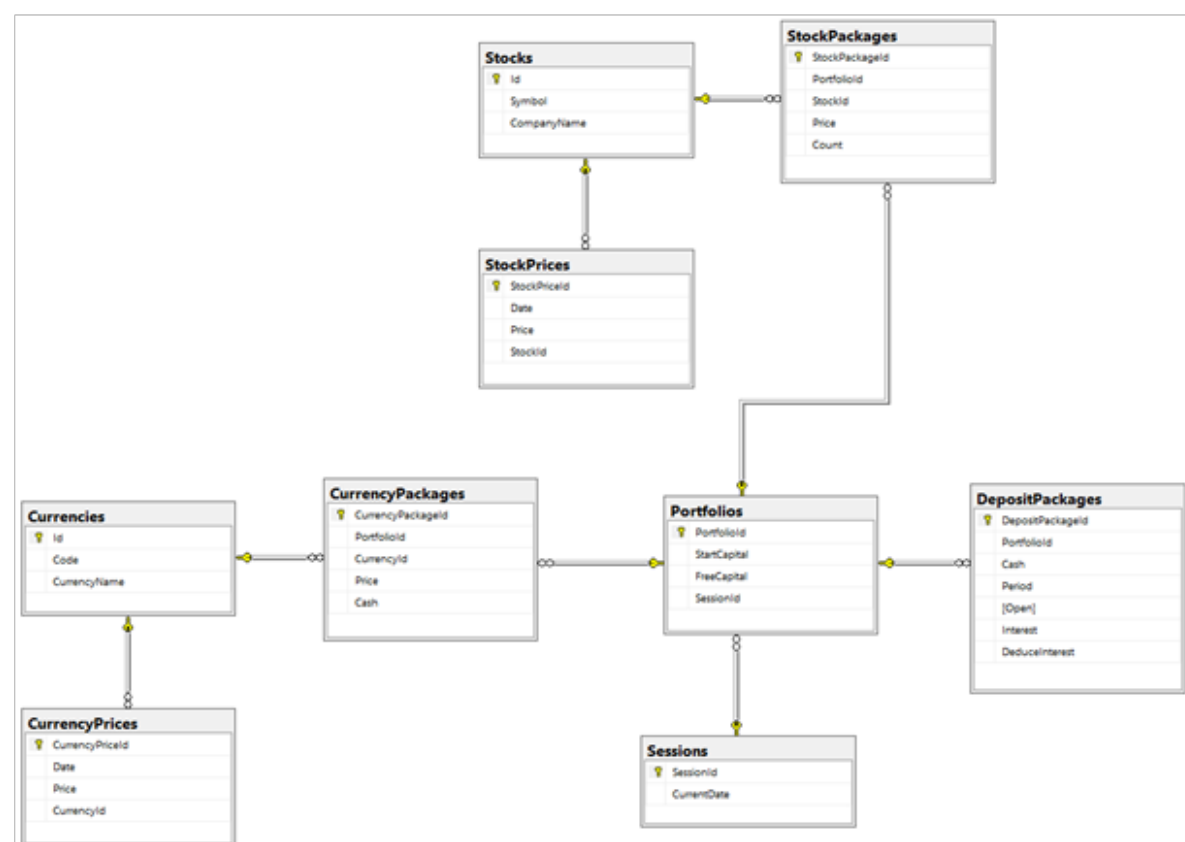
Валюта	Курсова ставка	Кількість акцій	Вартість акцій
Бит	EUR	1.1	1100
Оплатити	CHF	0.8	800

**Валюта**

Купити валюту з різних країн світу. Ви можете торгувати EUR, USD та інші валюти.

Ціна акцій компанії: 300  
Ваша ставка: 5000  
**Продати**

						ДП ІС-6127.06.000 КЕ			
Зм.	Арк.	№ документа	Підпис	Дата	Креслення екранних форм	Літера		Маса	Масштаб
Розробив		Ходарченко А.А							
Перевірив		Ковтунець О.В.							
Т. кон.									
						Аркуш 1		Аркушів 1	
Н. кон.		Телишева Т.О.			Інформаційна система з підтримки інвестиційно-фінансової діяльності	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-41			
Затвердив		Ковтунець О.В.							



ДП ІС-6127.05.000 СБД

Схема бази даних

Інформаційна система з підтримки  
інвестиційно-фінансової діяльності

Літера

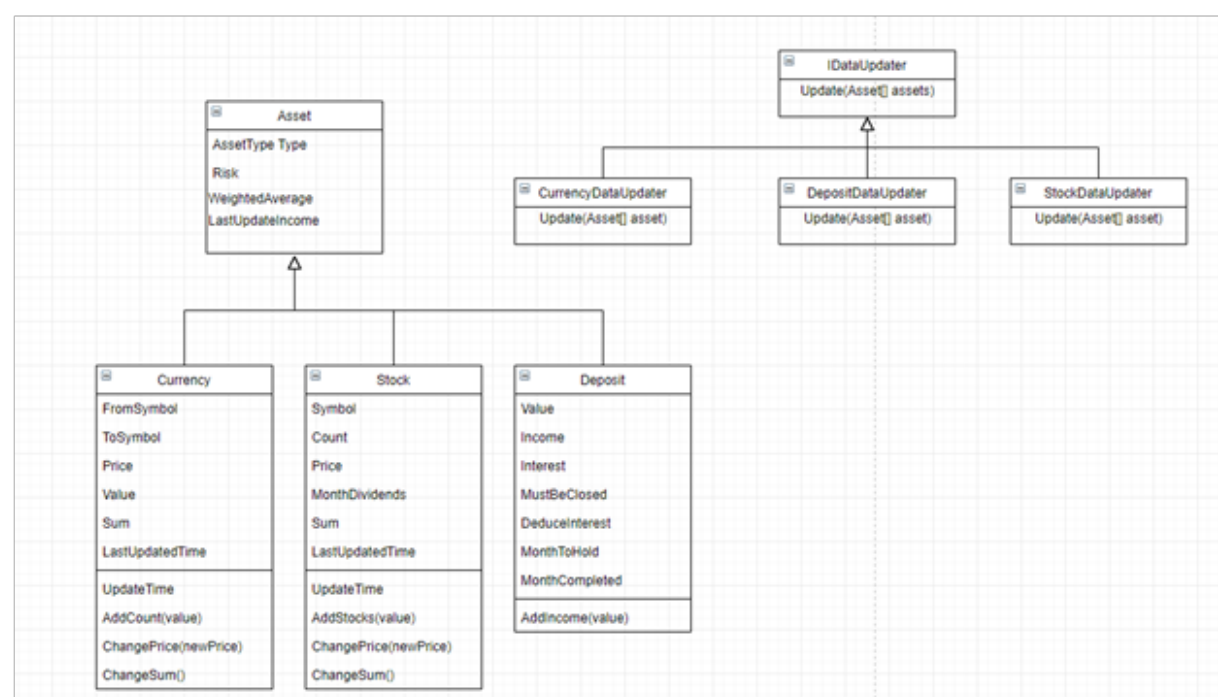
Маса

Масштаб

Аркуш 1

Аркушів 1

КПІ ім. Ігоря Сікорського  
кафедра АСОІУ гр. ІС-61



ДП ІС-6127.03.000 ССК

Схема структурна класів  
програмного забезпечення

Літера

Маса

Масштаб

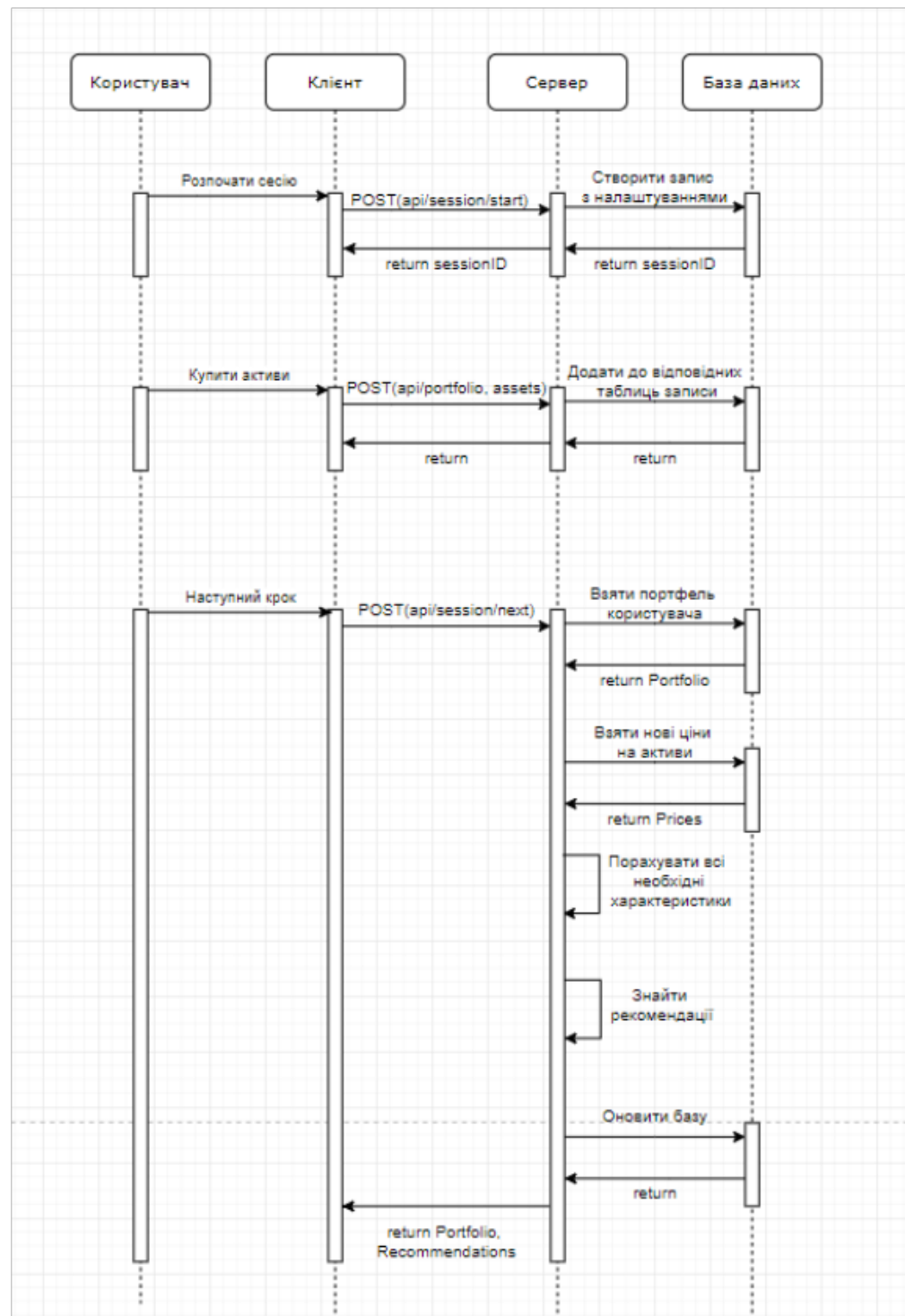
Зм.	Арк.	№ документа	Підпис	Дата
Розробив		Ходарченко А.А.		
Перевірив		Ковтунець О.В.		
Т. кон.				
Н. кон.		Телишева Т.О.		
Затвердив		Ковтунець О.В.		

Інформаційна система з підтримки  
інвестиційно-фінансової діяльності

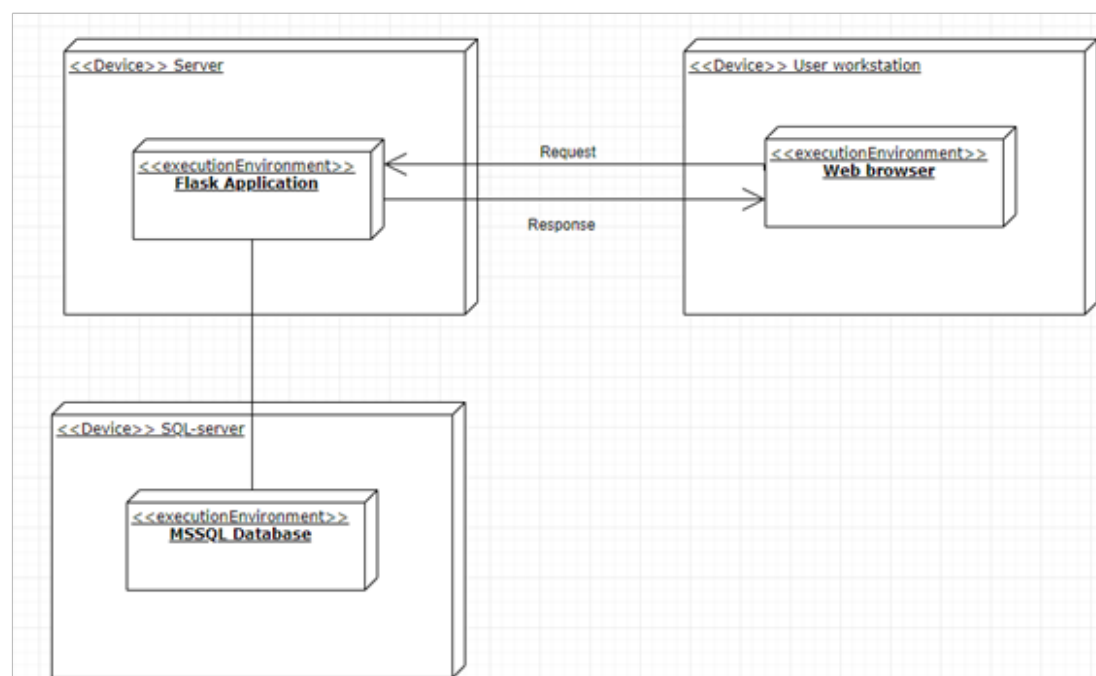
Аркуш 1

Аркушів 1

КПІ ім. Ігоря Сікорського  
кафедра АСОІУ гр. ІС-61



					ДП ІС-6127.02.000 ССП				
					Структурна схема послідовності	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив	Ходарченко А.А.								
Перевірив	Ковтунець О.В.								
Т. кон.						Аркуш 1		Аркушів 1	
Н. кон.	Телишева Т.О.				Назва дипл. проєкту	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61			
Затвердив	Ковтунець О.В.								



ДП ІС-6127.04.000 ССР

Структурна схема розгортання

Інформаційна система з підтримки  
інвестиційно-фінансової діяльності

Літера

Маса

Масштаб

Аркуш 1

Аркушів 1

КПІ ім. Ігоря Сікорського  
кафедра АСОІУ гр. ІС-61